File updates under random/arbitrary insertions and deletions

File Updates Under Random/Arbitrary Insertions And Deletions

Qiwen Wang, Viveck Cadambe, Sidharth Jaggi, Moshe Schwartz, and Muriel Médard

Abstract

A client/encoder edits a file, as modeled by an insertion-deletion *(InDel)* process. An old copy of the file is stored remotely at a data-centre/decoder, 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 server to update its copy to the newly edited file. We study two models for the source files/edit patterns: 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/deletions is a small (but constant) fraction of the original file. For both models we prove information-theoretic lower bounds on the best possible compression rates that enable file updates. Conversely, our compression algorithms use *dynamic programming (DP)* and entropy coding, and achieve rates that are approximately optimal.

I. Introduction

As the paradigm of cloud computing becomes pervasive, storing and transmitting files and their edited versions consumes a huge amount of resources (storage, bandwidth, computation) in client-datacentre channels, and intra-datacentre traffic. Industrial projections [1] predict the size of the digital universe will expand exponentially to 40 zetabytes (ZB) in 2020. By then, nearly 40 % of information will be "touched" by cloud computing [1].

If a file is "lightly edited", storing and transmitting the entire new file from clients to servers wastes a significant amount of space and bandwidth. Scenarios in which the number of edits is a small fraction of the original file are very common in real-life editing behaviour. 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 [2]–[4]. However, to the best of our knowledge, no existing techniques provide information-theoretically optimal compression guarantees, and indeed this is the primary contribution of our work.

There are potentially many other types of edits besides symbol insertions and deletions (for instance block insertions/deletion, substitutions, transpositions, copy-paste, crop, *etc.* – these and other edit models have been considered in, among other works, [5]–[10]). Since these other edit models are in general a combination of symbol insertions and deletions, we focus on the "base case" of symbol insertions-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) drawn from some distribution, and edits it according to some process – we shortly describe both the source and the edit process 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. [3]

We now discuss the pre-edit source sequence, and the edit process. There are many possible combinations of different pre-edit source sequence processes, and edit processes. Some of those that have been studied in the literature include: arbitrary input processes [9], [11], random input processes [10], [12]–[14], (partial) permutations [5], duplications [15]; random edit processes [9]–[13], Markov edit processes [14].

In this work, we consider two models. In the Random Pre-Edit Sequence, Left-to-Right Random InDel (RPES-LtRRID) process, a file is modeled as a sequence of symbols drawn i.i.d. uniformly at random from an alphabet $\mathcal{A}$. The new file is obtained from the old file through a *left-to-right random InDel process*, which is modeled as a Markov chain of three states: the "insert symbol" state, the "delete symbol" state, and the "no-operation" state. Roughly speaking, these three states correspond

[1] A caveat here – 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 practice.

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

[3] Several authors have considered the "interactive communication" version of the problem, in which the encoder and decoder communicate in multiple rounds. While tis is an interesting problem in its own right, we choose to focus on the relating less explored one-way communication problem, since as we show, there is little throughput penalty with such a restriction.

Ref	[1]Prob Description	[2] $\mathcal{A}$ size	[3]Pre-ESS	[4]Edits	[5]Edit Ops	[6] #(Edits) as $(\epsilon+\delta)n$	[7] Explicit Info Theo LB	[8]Algo	[9]Comp	[10]P_e	[11] #(bits) transmitted	[12] Remarks
[17]O93	$Enc \Leftarrow \mathbf{X}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightleftharpoons Dec$	2	Arb	Arb	Ins,Del,etc.	$\mathcal{O}(n)$	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	$\mathcal{O}(n^2)$		$(\epsilon+\delta)n[\log((1+\epsilon+\delta)n)+2]$	
[7]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 $(\epsilon+\delta)n\log n$
[6]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
[11]VZR10	$Enc \Leftarrow \mathbf{X}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightleftharpoons Dec$	2 / $\lvert\mathcal{A}\rvert$	Arb	Ran	Ins,Del	$o(\frac{n}{\log n})$	-	D	$\mathcal{O}(n)$	ε	$\mathcal{O}((\epsilon_n+\delta_n)n\log n)$ / $\mathcal{O}((\log\lvert\mathcal{A}\rvert)(\epsilon_n+\delta_n)n\log n)$	build on VT code [18]
[9]VSR13	$Enc \Leftarrow \mathbf{X}$ $Dec \Leftarrow \mathbf{Y}$ $Enc \rightleftharpoons Dec$	2 / $\lvert\mathcal{A}\rvert$	Arb	Ran	Ins,Del,Sub	$o(n)$	-	D	$\mathcal{O}(n)$	ε	$\Theta((\epsilon_n+\delta_n)^{2/3}n\log n)$ / $\Theta((\log\lvert\mathcal{A}\rvert)(\epsilon_n+\delta_n)^{2/3}n\log n)$	
[12]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	$\mathcal{O}((-\delta\log\delta)n)$	
[13]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
[14]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
[10]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	$(\lim_{n\to\infty} H(\mathbf{X}\mid\mathbf{Y})/n+$ $\mathcal{O}(\max(\epsilon,\delta)^{2-\tau}))n$	
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	$(H(\delta)+H(\epsilon)+\epsilon\log\lvert\mathcal{A}\rvert$ $+\overline{\lambda}_A\cdot\epsilon^2)n$	
			Ran	Ran	Ins,Del	$\mathcal{O}(n)$	Y	D	$\mathcal{O}(n^2)$	ε	$(H(\delta)+H(\epsilon)+\epsilon\log\lvert\mathcal{A}\rvert$ $+\overline{\lambda}_R\cdot\max(\epsilon,\delta)^{2-\tau(\epsilon,\delta)})n$	

Table 1: (Related work) 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] '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. [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 [9], [11]–[13], the fractions of insertions and deletions vanish with n, hence the corresponding variables are denoted ϵ_n and δ_n. [12] This column has additional remarks on specific works.

to the cursor moving "from left to right", and at each point, either a uniformly random symbol is inserted, the symbol at the cursor is deleted, or the cursor jumps ahead without changing the previous symbol. This model attempts to capture a "one-pass/streaming" edit process.[4]

We also study an Arbitrary Pre-Edit Sequence, Arbitrary InDel (APES-AID) process. In this model, the old file is modeled as an arbitrary sequence over an arbitrary alphabet $\mathcal{A}$. The post-edit source sequence $\mathbf{Y}$ is generated from the pre-edit source sequence $\mathbf{X}$ through an *arbitrary/"worst-case" InDel process* – we require that the number of edit operations is at most a small (but possibly constant) fraction of the file length n. The sequence of edits (insertions and deletions) is arbitrary up to an upper bound on the total number, occurs in arbitrary positions, and inserts arbitrary symbols from $\mathcal{A}$ for edits corresponding to insertions. Both these models are described formally in Section II-B.

In both our models, we consider arbitrary alphabet sizes. We first prove information-theoretic lower bounds on the compression rate needed so that the decoder is able to reconstruct $\mathbf{Y}$ for both models. To do so we build non-trivially on recent work on the deletion channel [16] in the random pre-edit sequence/edit model (see Theorem 8), and provide a combinatorial argument in the arbitrary pre-edit source/edit model (see Theorem 9). We then design "universal" computationally-efficient achievability schemes based on dynamic programming (DP) and entropy coding (see Theorems 10 & 11). The compression rate achieved by the DP scheme is an explicitly computable additive term away from the lower bound for almost all alphabet-sizes[5], and number of edits. In the regime wherein the number of edits is a small (but possibly constant) fraction of the length of $\mathbf{X}$ and the alphabet size is large, this term is small (details in Section IV-B).

[4]More general/realistic sources/Markov edit-processes are the subject of our ongoing research.

[5]In the random source/edit model, we actually have no restriction on the alphabet-size; in the arbitrary source/edit model, for technical reasons, our bounds hold only for alphabets of size at least 3.

B. Related work

Various models of the file-synchronization problem have been considered in the literature – see Table 1 for a summary. Our work here differs from each of those 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 [6], [7], [9], [11]–[13]); hence our protocols are information-theoretically near-optimal (however for two-way communication model, computationally efficient schemes which achieve rates with constant factors to the lower bounds are already challenging). The one-way communication model studied in [10], [14] is the closest to our RPES-LtRRID model. For the information-theoretical lower bound, we differ from [14] by considering both insertions and deletions, and arbitrary alphabet. The achievability scheme in [10] matches the lower bound up to first order term for the random source/edit model, whereas our scheme is "universal" for both RPES-LtRRID and APES-AID models in our work. The literature on insertion/deletion channels and error-correcting codes is also quite closely related – indeed, we borrow significantly from techniques in [16], [19], [20].

There are two lines of related work. In file synchronization problem, the encoder knows $\mathbf{X}$ and the decoder knows $\mathbf{Y}$. The purpose is to let the decoder learn $\mathbf{X}$ (the encoder may or may not learn $\mathbf{Y}$) through communication (either two-way or one-way). In our file update problem, the encoder knows both $\mathbf{X}$ and $\mathbf{Y}$, the decoder knows $\mathbf{X}$. The purpose is to let the decoder learn $\mathbf{Y}$ by one-way communication. In [9], an interactive synchronization algorithm was introduced which corrects $o(n)$ random insertions, deletions and substitutions in binary alphabet, where n represents the file size. This is adapted from their previous work [11] which corrects $o(n/\log n)$ insertions and deletions. Their algorithm was used as a component in [12] where the synchronization algorithm corrects a small constant fraction of deletions over the binary alphabet, and in [13] wherein the algorithm synchronized insertions and deletions under non-binary non-uniform source. A one-way file synchronization model was studied in [14] with Markov deletions in binary alphabet, in which an optimal rate in an information theoretic expression was proved. In [10], a one-way file synchronization algorithm was introduced (with both versions available at the encoder) that synchronizes random insertions, deletions and substitutions over the binary alphabet.

In the insertion/deletion channel problem, the channel model there can be the same as our InDel process (there are many different ways to model the stochastic insertions/deletions in both problems). The purposes are different. In insertion/deletion channels, one need to choose the input distribution to maximize the channel capacity $\max_{p(\mathbf{X})} I(\mathbf{X};\mathbf{Y}) = \max_{p(\mathbf{X})} H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{X})$. In file updating problem, the input distribution is given (arbitrary and random in this paper). The purpose is to find the minimum amount of information Enc need to send to Dec $\min_{p(\mathbf{Y}|\mathbf{X})} H(\mathbf{Y}|\mathbf{X})$, where the probability $p(\mathbf{Y}|\mathbf{X})$ is determined by the InDel process.

II. Model

A. Notational Convention

In this work, our notational conventions are as follows. We denote scalars by lowercase nonboldface nonitalic symbols such as c. We use uppercase nonboldface symbols such as X to denote random variables, and lowercase nonboldface symbols such as x to denote instantiations of those random variables. We denote vectors (sequences) of random variables or their instantiations by boldface symbols, for example, $\mathbf{X}$ and $\mathbf{x}$ are vectors of random variable X and its instantiations x respectively. We also denote matrices by uppercase boldface symbols. For example, an m by n matrix is denoted by $\mathbf{M}_{m\times n}$, and when there is no ambiguity we abbreviate it by dropping the dimensions, such as $\mathbf{M}$. An n by n identity matrix is denoted by $\mathbf{I}_n$. We denote sets by calligraphic symbols, such as $\mathcal{S}$. The length of a vector $\mathbf{X}$ is denoted by $|\mathbf{X}|$. The cardinality of a set $\mathcal{S}$ is denoted by $|\mathcal{S}|$. We denote standard binary entropy by $H(\cdot)$, that is, $H(p) = -p\log p - (1-p)\log(1-p)$. All logorithms are binary.

B. Edit Process

1) Random Pre-Edit Sequence Left-to-Right Random InDel (RPES-LtRRID) Process: As noted in the introduction, many different stochastic models for source sequences and edit processes have been considered in the literature. In this work, we study a *RPES-LtRRID* process as shown in Fig. 1, which is motivated by the Markov deletion model in [14]. It is an i.i.d. insertion-deletion process, a special case of a more general left-to-right Markov InDel process as shown in Fig. 2. Our results should in general translate over to other stochastic models as well in the regime wherein there are a small number of insertions and deletions. But for the sake of concreteness, we focus on the i.i.d. left-to-right random InDel process.

- *Pre-edit source sequence (PreESS):* The source initially has a *pre-edit source sequence* $\bar{\mathbf{X}} = (\bar{X}_1, \bar{X}_2, \dots, \bar{X}_n)$, a length-$n$ sequence of symbols drawn i.i.d. uniformly at random from the source alphabet $\mathcal{A} = \{0, \dots, a-1\}$. Finally, we append an *end of file* symbol $\bar{X}_{n+1} = \mathsf{eof}$ to the end of $\bar{\mathbf{X}}$. We denote the distribution of the pre-edit source sequence by $p(\bar{\mathbf{X}})$.
- *InDel process:* As shown in Fig. 1, the InDel process is a Markov Chain with three states as defined in the following:
  - the "insertion state" $\bar{\iota}$: insert (write) a symbol uniformly drawn from $\mathcal{A}$;
  - the "deletion state" $\bar{\Delta}$: read one symbol rightwards in the pre-edit source sequence $\bar{\mathbf{X}}$, and delete the symbol;
  - the "no-operation state" $\bar{\eta}$: read one symbol rightwards in the pre-edit source sequence $\bar{\mathbf{X}}$, and do nothing.

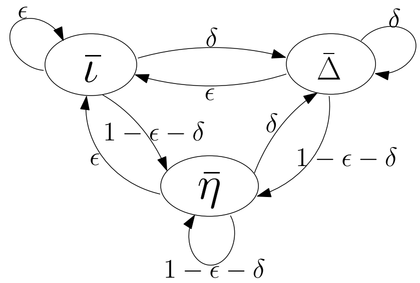

Fig. 1: *Left-to-Right Random InDel (LtRRID)* process: Starting in front of the first symbol of $\bar{\mathbf{X}}$, at each step, the process inserts a symbol uniformly drawn from $\mathcal{A}$ with probability ϵ, reads one symbol rightwards and deletes it with probability δ, reads one symbol rightwards and does nothing with probability $1-\epsilon-\delta$. Note that an inserted symbol is never deleted in this process. In contrast, a deleted symbol might be inserted back right away, with probability $\epsilon\frac{1}{|\mathcal{A}|}$. The process stops when it reaches the end of file $\bar{X}_{n+1} = \text{eof}$.

The edit process starts in front of $\bar{X}_1$ and ends when it reaches the end of file $\bar{X}_{n+1} = \text{eof}$. This means that in our model, the total number of deletions plus no-operations equals exactly n. In addition there are a potentially unbounded number of insertions (though in our model the expected number of insertions in bounded).[6] The number of deletions and insertions are random variables K_D and K_I respectively. We describe the *edit pattern* of the InDel process by a pair of sequences $\bar{\mathbf{E}} = (\bar{O}^{n+K_I}, \bar{C}^{K_I})$, where the edit operation pattern is $\bar{O}^{n+K_I} \in \{\bar{\iota}, \bar{\Delta}, \bar{\eta}\}^{n+K_I}$ and the insertion content is $\bar{C}^{K_I} \in \mathcal{A}^{K_I}$. The *random* (ϵ, δ)*-InDel process* is an i.i.d. insertion-deletion process with $P(\bar{\iota}) = \epsilon$, $P(\bar{\Delta}) = \delta$, and $P(\bar{\eta}) = 1-\epsilon-\delta$.

- *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 $\bar{\mathbf{X}}$ through the InDel process $\bar{\mathbf{E}} = (\bar{O}^{n+K_I}, \bar{C}^{K_I})$.
- *Post-edit set:* Given any PreESS $\bar{\mathbf{X}}$, any PosESS $\bar{\mathbf{Y}}$ in $\mathcal{A}^*$ (any sequence over $\mathcal{A}$ of any length) might be in its post-edit set, albeit with possibly "very small" probability. In fact, for any $\bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}$, there may be multiple edit patterns that generate $\bar{\mathbf{Y}}$ from $\bar{\mathbf{X}}$. We use $p(\bar{\mathbf{Y}}|\bar{\mathbf{X}})$ to denote the probability that the output of the random left-to-right InDel process generates $\bar{\mathbf{Y}}$ from $\bar{\mathbf{X}}$ (via any edit pattern).
- *Runs:* We use the usual definition (see, for example [21]) of a *run* being a maximal block of contiguous identical symbols. Since we shall be interested in runs of several different sequences, to avoid confusion about the parent sequence we use $\mathbf{S}$*-run* to denote a run in a sequence $\mathbf{S}$.

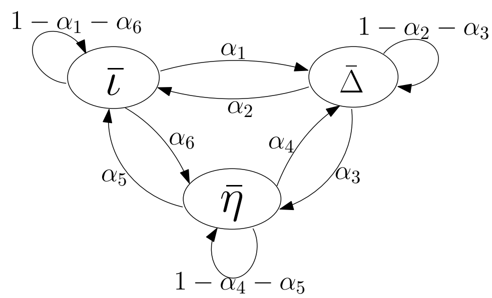

Fig. 2: *General Left-to-Right Markov InDel (GLtRMID)* process: a general three-state Markov Chain where transitions between any of the three states can happen with general probabilities. This results in an InDel process with unit memory. However, the block lengths of insertions and deletions are still geometrically distributed. This model is a subject of our ongoing research.

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

- *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 in $\mathcal{A}^n$.
- *InDel process:* The *InDel process* consists of a sequence of *arbitrary InDel edits* $\mathbf{E} = (E_1, E_2, \ldots, E_k)$, where k denotes the *number of edits*. For notational convenience we also use $\mathbf{X}_0$ to denote $\mathbf{X}$, and $\mathbf{X}_j$ to denote the sequence obtained from $\mathbf{X}_0$ after the first j edits $(E_1, \ldots, E_j)$ for all $j = 1, 2, \ldots, k$. An *arbitrary InDel edit* $E_j = (P_j, O_j, C_j)$ consists of three parameters:

[6]Note that in our model a symbol that is inserted cannot be deleted, since the "cursor" moves on after inserting a symbol. This is just one of many possible stochastic InDel processes – we choose to work with this model since it makes notation more convenient – we believe similar results can be obtained for a variety of related stochastic InDel processes.

- the position of the *cursor* $P_j \in \{0, 1, 2, \ldots, |\mathbf{X}_{j-1}|\}$, which is the positions between symbols (including in front of the first symbol and behind the last symbol) in the *current sequence* $\mathbf{X}_{j-1}$;
- the *edit operation* $O_j \in \{\iota, \Delta\}$, where ι indicates that the edit operation is inserting at the cursor position, and Δ indicates that the edit operation is deleting the symbol in front of the cursor (when $P_j = 0$, the edit operation can only be an insertion, that is, $O_j = \iota$);
- the content of insertion $C_j \in \mathcal{A} \cup \{\text{nop}\}$, which is an arbitrary symbol from $\mathcal{A}$ if the edit operation is an insertion, and "nop" if the edit operation is a deletion.

The sequence obtained from $\mathbf{X}_{j-1}$ after the jth arbitrary InDel edit E_j is a function of $\mathbf{X}_{j-1}$ and E_j, and is denoted by $\mathbf{X}_j = \mathbf{X}_j(\mathbf{X}_{j-1}, E_j)$. The edit process defined as above is an *arbitrary InDel process*. If the edit process subjects to the constraint that there are at most ϵn insertions and δn deletions, it is called an *arbitrary* (ϵ, δ)*-InDel process*. (Since the sequence length keeps changing, for clarity, the parameters are with respect to the length of the pre-edit source sequence.) Two special cases are the *arbitrary* ϵ*-insertion process* (equivalently an arbitrary $(\epsilon, 0)$-InDel process), and the *arbitrary* δ*-deletion process* (equivalently an arbitrary $(0, \delta)$-InDel process).

- *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 $\mathbf{X}$ through an arbitrary InDel process $\mathbf{E} = \{E_1, \ldots, E_k\}$. If the InDel process is subject to an (ϵ, δ)-constraint, the post-edit source sequence is called an (ϵ, δ)*-post-edit source sequence*.
- $(\mathbf{X}, (\epsilon, \delta))$*-post-edit set:* Let $\mathcal{Y}_{\epsilon,\delta}(\mathbf{X})$ denote the $(\mathbf{X}, (\epsilon, \delta))$*-post-edit set* – the set of all sequences over $\mathcal{A}$ that may be obtained from $\mathbf{X}$ via the arbitrary (ϵ, δ)-InDel process.
- *Runs:* The same as defined in the RPES-LtRRID model, a *run* is a maximal block of contiguous identical symbols.

Remark: Note that in the *APES-AID process*, the order of insertions and deletions in the edit process is in general arbitrary. However, based on the following Fact 1, we can simplify the model by separating the insertions and deletions.

Fact 1. *An arbitrary (ϵ, δ)-InDel process can be separated to an arbitrary δ-deletion process followed by an arbitrary $\frac{\epsilon}{1-\delta}$-insertion process.*

The proof of Fact 1 is provided in Appendix B.

C. Communication Model

The communication system is as shown in Fig. 3. We define the communication model for both RPES-LtRRID process and APES-AID process. For clarity, we state the model for the RPES-LtRRID process, and repeat for the APES-AID process using notation without bars.

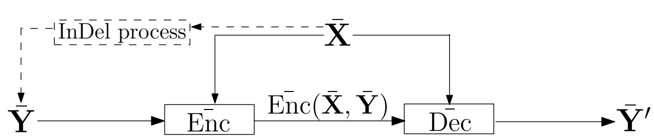

Fig. 3: Communication model: The source has both the random PreESS $\bar{\mathbf{X}}$ and the random PosESS $\bar{\mathbf{Y}}$, as discussed in Section II-B1. The sequence $\bar{\mathbf{Y}}$ is obtained from $\bar{\mathbf{X}}$ through the random (ϵ, δ)-InDel process discussed in Section II-B1. The source encodes the source sequences $(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ into a transmission $\bar{\mathrm{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ and sends it to the decoder through a noiseless channel. The arbitrary PreESS $\bar{\mathbf{X}}$ is available at the decoder as side-information. The decoder receives $\bar{\mathrm{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$, and regenerates the arbitrary PosESS $\bar{\mathbf{Y}}'$ from $(\bar{\mathrm{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}}), \bar{\mathbf{X}})$. Here the bar superscript is used to denote the fact that the source sequences and edit process are as described in Section II-B1 rather than Section II-B2. The communication model for the APES-AID model discussed in Section II-B2 is similar, except that the quantity $\{\bar{\mathbf{X}}, \bar{\mathbf{Y}}, \bar{\mathrm{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}}), \bar{\mathbf{Y}}'\}$ are replaced with $\{\mathbf{X}, \mathbf{Y}, \mathrm{Enc}(\mathbf{X}, \mathbf{Y}), \mathbf{Y}'\}$.

In the *RPES-LtRRID process* 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 process. The PreESS $\bar{\mathbf{X}}$ and PosESS $\bar{\mathbf{Y}}$ are encoded using an *encoder* $\bar{\mathrm{Enc}}$. Its output is possibly any non-negative integer $\bar{\mathrm{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$. Taking as inputs the transmission $\bar{\mathrm{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ and the PreESS $\bar{\mathbf{X}}$, the *decoder* Dec reconstructs the PosESS $\bar{\mathbf{Y}}$ as $\bar{\mathbf{Y}}'$. The code $\bar{\mathcal{C}}_n^{\epsilon,\delta}$ comprises the encoder-decoder pair $(\mathrm{Enc}, \mathrm{Dec})$. The *average rate* $\bar{R}$ of the code $\bar{\mathcal{C}}_n^{\epsilon,\delta}$ is the average number of bits transmitted by the encoder, defined as $\sum_{\bar{\mathbf{X}} \in \mathcal{A}^n, \bar{\mathbf{Y}} \in \mathcal{A}^*} p(\bar{\mathbf{X}}, \bar{\mathbf{Y}}) \log |\bar{\mathrm{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})|$. A code $\bar{\mathcal{C}}_n^{\epsilon,\delta}$ is "$(1-P_e)$-good" if the *average probability of error*, defined as $\Pr_{\bar{\mathbf{X}} \in \mathcal{A}^n, \bar{\mathbf{Y}} \in \mathcal{A}^*}\{(\bar{\mathbf{X}}, \bar{\mathbf{Y}}) : \bar{\mathrm{Dec}}(\bar{\mathrm{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}}), \bar{\mathbf{X}}) \neq \bar{\mathbf{Y}}\}$, is less than P_e. A rate $\bar{R}_{\epsilon,\delta}$ is said to be *achievable on average* if for any $P_e > 0$ there is a code for sufficiently large n such that it is $(1 - P_e)$-good. The infimum over (over all n and corresponding $\bar{\mathcal{C}}_n^{\epsilon,\delta}$) of all achievable rates is called the *optimal average transmission rate*, and is denoted $\bar{R}^*_{\epsilon,\delta}$.

In the *APES-AID process* 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 process. The PreESS $\mathbf{X}$ and PosESS $\mathbf{Y}$ are encoded using an *encoder* Enc into a *transmission* $\mathrm{Enc}(\mathbf{X}, \mathbf{Y})$ from the set $\{1, 2, \ldots, 2^{nR}\}$, where R denotes the *rate* of the encoder Enc. Taking as inputs the transmission $\mathrm{Enc}(\mathbf{X}, \mathbf{Y})$ 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 $(\mathrm{Enc}, \mathrm{Dec})$. A code $\mathcal{C}_n^{\epsilon,\delta}$ is said to be "good" if for every $\mathbf{X}$ in $\mathcal{A}^n$ and $\mathbf{Y}$ in the $(\mathbf{X}, (\epsilon, \delta))$-post-edit set, the decoder outputs the correct PosESS, *i.e.* $\mathbf{Y}' = \mathbf{Y}$. A rate $R_{\epsilon,\delta}$ is said to be *achievable* if for sufficiently large n there exists a good code with rate at most $R_{\epsilon,\delta}$. The infimum (over all n and corresponding $\mathcal{C}_n^{\epsilon,\delta}$) of all achievable rates is called the *optimal transmission rate*, and is denoted $R^*_{\epsilon,\delta}$.

Remark: For the *APES-AID process*, we require zero-error for the source code. Because we can achieve this stringent requirement without paying a penalty in our optimal achievable rate. Conversely, we allow "small" error in the *RPES-LtRRID process*. Because it is necessary to allow for "atypical" source sequences and edit patterns.

III. Lower Bound

A. RPES-LtRRID Process

1) Proof Roadmap: Since the decoder already has access to the PreESS $\bar{\mathbf{X}}$, the entropy of $\bar{\text{Enc}}(\bar{\mathbf{X}},\bar{\mathbf{Y}})$ 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 "complicated" function of n – indeed the same challenge is faced in providing information-theoretic converses for *any* 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 update problem. We follow the lead of [16], 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, and insertions and their contents (this is argued formally in Lemma 4 below). [7] Since multiple edit patterns can take a PreESS $\bar{\mathbf{X}}$ to a 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 [14]. For instance, given $\bar{\mathbf{X}} = 00000$ and $\bar{\mathbf{Y}} = 000$, the decoder doesn't 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 doesn't 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 "long sequences with memory". One challenge is that it is hard to align $\bar{\mathbf{X}}$-runs and $\bar{\mathbf{Y}}$-runs. In other words, it's in general difficult to tell which run/runs in $\bar{\mathbf{X}}$ lead to a run in $\bar{\mathbf{Y}}$ (we call this run/runs in $\bar{\mathbf{X}}$ the *parent run/runs* of the run in $\bar{\mathbf{Y}}$ [16]). We develop the approach in [16]:

- We first carefully "perturb" the original edit pattern $\bar{\mathbf{E}}$ to a *typicalized edit pattern* $\hat{\mathbf{E}}$ (described in details below).
- We compute the *typicalized PosESS* $\hat{\mathbf{Y}}$ corresponding to operating the typicalized edit pattern $\hat{\mathbf{E}}$ on the PreESS $\bar{\mathbf{X}}$.
- We show via non-trivial case analysis and Lemma 6 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 $H(\hat{\mathbf{E}}|\bar{\mathbf{X}},\hat{\mathbf{Y}})$, and 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. 4.

$$(\bar{\mathbf{E}},\bar{\mathbf{X}},\bar{\mathbf{Y}}) \xleftrightarrow{\hat{\mathbf{E}}^C} (\hat{\mathbf{E}},\bar{\mathbf{X}},\hat{\mathbf{Y}})$$

$$(\bar{\mathbf{X}},\bar{\mathbf{Y}}) \xleftrightarrow{\hat{\mathbf{E}}^C,\, A_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}} (\bar{\mathbf{X}},\hat{\mathbf{Y}})$$

$$n\bar{R}_{\epsilon,\delta} \overset{\text{Lemma 2 (Fano's)}}{\geq} H(\bar{\mathbf{Y}}|\bar{\mathbf{X}}) \overset{\text{Lemma 3}}{=} \overset{\text{Computed in Lemma 4}}{H(\bar{\mathbf{E}})} - H(\bar{\mathbf{E}}|\bar{\mathbf{X}},\bar{\mathbf{Y}}) = H(\bar{\mathbf{E}}) - \overset{\text{Bounded in Lemma 6}}{H(\hat{\mathbf{E}}|\bar{\mathbf{X}},\hat{\mathbf{Y}})} + \overbrace{H(\hat{\mathbf{E}}|\bar{\mathbf{X}},\hat{\mathbf{Y}}) - H(\bar{\mathbf{E}}|\bar{\mathbf{X}},\bar{\mathbf{Y}})}^{\text{Bounded in Lemma 7}} \geq H(\bar{\mathbf{E}}) - H(\hat{\mathbf{E}}|\bar{\mathbf{X}},\hat{\mathbf{Y}}) - 2H(\hat{\mathbf{E}}^C) - H(A_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})$$

Fig. 4: 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 $H(\bar{\mathbf{E}})$ subtracts 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. 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 $\hat{\mathbf{E}}$. 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 $\mathcal{O}(\max(\epsilon,\delta)^2)$ distance from the "typicalized nature's secret" $H(\hat{\mathbf{E}}|\bar{\mathbf{X}},\hat{\mathbf{Y}})$, which we characterize in Lemma 6.

One major difference between our work and the analysis in [16] 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

[7] 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/smaller" term than the terms we focus on in this work, in the regime of small ϵ,δ.

all (finite) alphabet sizes, whereas [16] concerned itself only with binary sequences. Also, besides the difference in models and techniques, the underlying motivation differs. The authors of [16] focused on characterizing the capacity of deletion channels (and hence they could choose arbitrary subsets of PreESS). On the other hand we focus on the file update problem (and hence our "channel input" PreESS $\bar{\mathbf{X}}$ is drawn according to source statistics).

2) Proof Details: Recall in the InDel model (described in Section II-B1), 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_D 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_I 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 $\mathrm{Geo}_0(1-\epsilon)$, the geometric distribution on the support of $\{0, 1, 2, \ldots\}$ with parameter $(1-\epsilon)$ [22]. The InDel process stops when the total number of deletions and no-operations is n. Hence, K_I is the sum of $n+1$ i.i.d. random variables whose distributions follow $\mathrm{Geo}_0(1-\epsilon)$. On the other hand, K_I is the number of insertions with probability ϵ until $n+1$ deletions/no-operations occur, which follows a negative binomial distribution $\mathrm{NB}(n+1; \epsilon)$ with mean $(n+1)\frac{\epsilon}{1-\epsilon}$ [22].

Throughout this section, because we deal with sequences with random lengths, we use Theorem 3 in [23] multiply times. Hence we restate the theorem here as a preliminary for our later proofs.

Theorem 1. *[23] [Theorem 3 (Determined Stopping Time)] A stopping time N is said to be a determined stopping time for the i.i.d. sequence $X_1, X_2, \ldots$ if $\{N = n\} \in \sigma(X_1, X_2, \ldots, X_n)$ for all $n = 1, 2, \ldots$, where $\sigma(X_1, X_2, \ldots, X_n)$ is the σ-field generated by $X_1, X_2, \ldots, X_n$. Then, for a determined stopping time N,*

$$H(X^N) = E[N]H(X_1), \tag{1}$$

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

Lemma 2 (Converse)**.** *For the Random Pre-Edit Sequence Left-to-Right Random InDel (RPES-LtRRID) process, the achievable rate $\bar{R}_{\epsilon,\delta}$ is at least $H(\bar{\mathbf{Y}}|\bar{\mathbf{X}})$.*

Proof: We firstly show a modified version of the conventional Fano's inequality $H(\bar{\mathbf{Y}}|\bar{\mathbf{Y}}') \le 1 + P_e \log|\bar{\mathbf{Y}}|$. Because we allow insertions in our model, 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 $H(\bar{\mathbf{Y}}|\bar{\mathbf{Y}}', \bar{\mathbf{Y}}' \neq \bar{\mathbf{Y}}) \le \log|\bar{\mathbf{Y}}|$ in the proof of the conventional Fano's inequality doesn't work in our problem. We modify the Fano's inequality bound the term by $H(\bar{\mathbf{Y}}|\bar{\mathbf{Y}}', \bar{\mathbf{Y}}' \neq \bar{\mathbf{Y}}) \le H(\bar{\mathbf{Y}})$. The PosESS $\bar{\mathbf{Y}}$ is a sequence of symbols drawn uniformly i.i.d. from $\mathcal{A}$, where its length $(n - K_D + K_I)$ is a "determined stopping time" for the sequence. Hence by Theorem 1, $H(\bar{\mathbf{Y}}) = (n - E[K_D] + E[K_I]) \log|\mathcal{A}| = \left(\frac{1-\delta}{1-\epsilon}n + \frac{\epsilon}{1-\epsilon}\right) \log|\mathcal{A}|$. Hence, our modified Fano's inequality is

$$H(\bar{\mathbf{Y}}|\bar{\mathbf{X}}, \bar{\mathrm{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})) \le 1 + P_e\left(\frac{1-\delta}{1-\epsilon}n + \frac{\epsilon}{1-\epsilon}\right)\log|\mathcal{A}| \le n\sigma_n, \tag{2}$$

where $\sigma_n \to 0$ as $n \to \infty$.

We have the following chain of inequalities,

$$\begin{aligned} n\bar{R}_{\epsilon,\delta} &\ge H(\bar{\mathrm{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})) \\ &\ge H(\bar{\mathrm{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})|\bar{\mathbf{X}}) \\ &= H(\bar{\mathbf{Y}}|\bar{\mathbf{X}}) + H(\bar{\mathrm{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})|\bar{\mathbf{X}}, \bar{\mathbf{Y}}) - H(\bar{\mathbf{Y}}|\bar{\mathbf{X}}, \bar{\mathrm{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})) \\ &\overset{(a)}{=} H(\bar{\mathbf{Y}}|\bar{\mathbf{X}}) - H(\bar{\mathbf{Y}}|\bar{\mathbf{X}}, \bar{\mathrm{Enc}}(\bar{\mathbf{X}}, \bar{\mathbf{Y}})) \\ &\overset{(b)}{\ge} H(\bar{\mathbf{Y}}|\bar{\mathbf{X}}) - n\sigma_n, \end{aligned} \tag{3}$$

where equality (a) holds since standard arguments show that randomized encoders do not help. Inequality (b) follows from our modified Fano's inequality as shown in Equation 2.

Dividing both sides of Equation 3 by n deduce 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:

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

$$\stackrel{(c)}{=} H(\bar{\mathbf{E}}) - H(\bar{\mathbf{E}}|\bar{\mathbf{X}}, \bar{\mathbf{Y}}),$$

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

Lemma 4. $\lim_{n\to\infty} \frac{1}{n} H(\bar{\mathbf{E}}) \geq H(\delta) + H(\epsilon) + \epsilon \log|\mathcal{A}| + 2\min(\epsilon,\delta)^{2-\tau} + \mathcal{O}(\max(\epsilon,\delta)^2)$

Proof: Recall that $\bar{\mathbf{E}} = (\bar{O}^{n+K_I}, \bar{C}^{K_I})$, where $\bar{O}^{n+K_I}$ 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,

$$\begin{aligned}
H(\bar{O}_1) &= -\delta\log\delta - \epsilon\log\epsilon - (1-\epsilon-\delta)\log(1-\epsilon-\delta) \\
&= H(\delta) + H(\epsilon) + (1-\delta)\log(1-\delta) + (1-\epsilon)\log(1-\epsilon) - (1-\epsilon-\delta)\log(1-\epsilon-\delta) \\
&\stackrel{(a)}{=} H(\delta) + H(\epsilon) + (1-\delta)(\log e)(-\delta - \frac{\delta^2}{2} - \mathcal{O}(\delta^3)) + (1-\epsilon)(\log e)(-\epsilon - \frac{\epsilon^2}{2} - \mathcal{O}(\epsilon^3)) - \\
&\quad (1-\delta-\epsilon)(\log e)[-(\delta+\epsilon) - \frac{(\delta+\epsilon)^2}{2} - \mathcal{O}((\delta+\epsilon)^3)] \\
&= H(\delta) + H(\epsilon) - \epsilon\delta\log e + \mathcal{O}(\max(\epsilon,\delta)^3),
\end{aligned} \tag{4}$$

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

$$\begin{aligned}
\lim_{n\to\infty}\frac{1}{n}H(\bar{\mathbf{E}}) &= \lim_{n\to\infty}\frac{1}{n}[H(\bar{O}^{n+K_I}) + H(\bar{C}^{K_I}|\bar{O}^{n+K_I})] \\
&\stackrel{(a)}{=} \lim_{n\to\infty}\frac{1}{n}[(n+E[K_I])H(\bar{O}_1) + H(\bar{C}^{K_I}|\bar{O}^{n+K_I})] \\
&\stackrel{(b)}{=} \lim_{n\to\infty}\frac{1}{n}[(n+E[K_I])H(\bar{O}_1) + H(\bar{C}^{K_I}|K_I)] \\
&= \lim_{n\to\infty}\frac{1}{n}[(n+E[K_I])H(\bar{O}_1) + \sum_{k=0}^{\infty} H(\bar{C}^{K_I}|K_I = k)\Pr(K_I = k)] \\
&= \lim_{n\to\infty}\frac{1}{n}[(n+E[K_I])H(\bar{O}_1) + \sum_{k=0}^{\infty} H(C^k)\Pr(K_I = k)] \\
&= \lim_{n\to\infty}\frac{1}{n}[(n+E[K_I])H(\bar{O}_1) + \sum_{k=0}^{\infty} kH(C_1)\Pr(K_I = k)] \\
&= \lim_{n\to\infty}\frac{1}{n}[(n+E[K_I])H(\bar{O}_1) + H(C_1)\sum_{k=0}^{\infty} k\Pr(K_I = k)] \\
&\stackrel{(c)}{=} \lim_{n\to\infty}\frac{1}{n}[(n+E[K_I])H(\bar{O}_1) + E[K_I]H(C_1)] \\
&\stackrel{(d)}{=} \lim_{n\to\infty}\frac{1}{n}\left[\frac{n+\epsilon}{1-\epsilon}H(\bar{O}_1) + (n+1)\frac{\epsilon}{1-\epsilon}\log|\mathcal{A}|\right] \\
&= \frac{1}{1-\epsilon}(H(\bar{O}_1) + \epsilon\log|\mathcal{A}|) \\
&\stackrel{(e)}{=} \frac{1}{1-\epsilon}\left(H(\delta) + H(\epsilon) + \epsilon\log|\mathcal{A}| - \epsilon\delta\log e + \mathcal{O}(\max(\epsilon,\delta)^3)\right) \\
&\stackrel{(f)}{=} H(\delta) + H(\epsilon) + \epsilon\log|\mathcal{A}| - \epsilon\delta\log\delta - \epsilon^2\log\epsilon + (\log e + \log|\mathcal{A}|)\epsilon^2 + \mathcal{O}(\max(\epsilon,\delta)^3) \\
&\geq H(\delta) + H(\epsilon) + \epsilon\log|\mathcal{A}| + 2\min(\epsilon,\delta)^{2-\tau} + \mathcal{O}(\max(\epsilon,\delta)^2),
\end{aligned}$$

where equality (a) is because by Theorem 3 in [23], $n + K_I$ is a "determined stopping time" for the i.i.d. edit sequence $\bar{O}_1, \bar{O}_2, \ldots$, hence $H(\bar{O}^{n+K_I}) = (n + E[K_I])H(\bar{O}_1)$. Equality (b) is because given the edit operation sequence $\bar{O}^{n+K_I}$, the insertion content sequence $\bar{C}^{K_I}$ depends only on the number of insertions K_I.[8] From equality (b) to equality (c) is by expanding K_I and noting that $\bar{C}^{K_I}$ is a sequence of i.i.d. variables. Equality (d) is by Fact **??**(a) and noting that the content of insertions are uniformly drawn from the alphabet. Equality (e) is by Equation 4. Equality (f) is by taking the Taylor series expansion of $\frac{1}{1-\epsilon}$, $H(\delta)$ and $H(\epsilon)$. □

As discussed in Section III-A1 and Fig. 4, the next quantity we need to calculate/bound is the "nature's secret" $H(\bar{\mathbf{E}}|\bar{\mathbf{X}}, \bar{\mathbf{Y}})$ of the edit process. However, this quantity is in general difficult to calculate because $\bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}$ are unsynchronized. Hence

[8]Equivalently, $H(\bar{C}^{K_I}|\bar{O}^{n+K_I}) = H(\bar{C}^{K_I}|\bar{O}^{n+K_I}, K_I) = H(\bar{C}^{K_I}|K_I) + H(\bar{O}^{n+K_I}|\bar{C}^{K_I}, K_I) - H(\bar{O}^{n+K_I}|K_I) = H(\bar{C}^{K_I}|K_I)$.

we perturb the edit process $\bar{\mathbf{E}}$ to a "typicalized edit process" $\hat{\mathbf{E}}$, 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 process $\hat{\mathbf{E}}$ and some sequences that depend on $\hat{\mathbf{E}}$:

Definition 1 (Typicalized edit process)**.** *The typicalized edit pattern $\hat{\mathbf{E}}$ is determined from $(\bar{\mathbf{X}},\bar{\mathbf{E}})$ by choosing a subset of the edits in the original edit pattern $\bar{\mathbf{E}}$ in the following way. The extended run [16] 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.[9] 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 $\bar{\mathbf{E}}: 0\cancel{1}11\cancel{2}23$, there are two edits in the extended run 01112 of the second run 111, hence the edit in the first 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 $\hat{\mathbf{E}}: 0111\cancel{2}23$.
- An insertion that occurs at the boundary of two runs is contained in the extended runs of both the run at its left and the run at its right. If there is more than one edit in at least one of the extended runs it belongs to, the insertion is eliminated in $\hat{\mathbf{E}}$. For example, for $\bar{E}: 0111^{\downarrow 4}22\cancel{3}$, in the extended run 01112 there is only one edit – the insertion of 4 in front of the right-neighbour. However, in the extended run 1223 there are two edits, the insertion of 4 is eliminated in $\hat{\mathbf{E}}$. The last symbol 3 is the right-neighbour of the run 22, hence its deletion is not eliminated in $\hat{\mathbf{E}}$. The typicalized edit pattern in this example is $011122\cancel{3}$.

Denote the number of insertions and deletions in the typicalized edit process $\hat{\mathbf{E}}$ by $\hat{K}_I$ and $\hat{K}_D$ 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 $n+\hat{K}_I$.

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 $n-\hat{K}_D+\hat{K}_I$.*

Definition 3 (Complement of the typicalized edit process)**.** *The complement of the typicalized edit process $\hat{\mathbf{E}}^C = (\underline{\bar{O}}^{n+K_I-\hat{K}_I}, \underline{\bar{C}}^{K_I-\hat{K}_I})$ is defined to specify the eliminated edits, where $\underline{\bar{O}}^{n+K_I-\hat{K}_I} \in \{-,\underline{\iota},\underline{\Delta}\}^{n+K_I-\hat{K}_I}$ specifies the positions and operations of the eliminated edits and $\underline{\bar{C}}^{K_I-\hat{K}_I} \in \mathcal{A}^{K_I-\hat{K}_I}$ specifies the contents of eliminated insertions.*

Fig. 5 shows an example of all the sequences we define above. We will reuse this example later multiple times to explain different concepts. Fig. 6 shows the dependencies of all the sequences we define above, and some internal random variables we define and use in the later proofs.

We first show that $\hat{\mathbf{Y}}$-runs can be "mostly" aligned to the parent run/runs in $\bar{\mathbf{X}}$. The intuition is that since $\bar{\mathbf{X}}$-runs undergo at most one edit in the typicalized edit process, for any $\hat{\mathbf{Y}}$-run, there are only a few possible cases for its parent run(runs), and the corresponding length(lengths). There are only two events where the cases of the parent run-length intersect, which we call the "ambiguous local alignment" events. An ambiguous local alignment event might be resolved by keeping aligning both possible alignments, until for one alignment no typicalized edit pattern can convert $\bar{\mathbf{X}}$ to $\hat{\mathbf{Y}}$. Otherwise, both local alignments are possible and results in different "global alignments". 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. Fig. 8 gives an example where an ambiguous local alignment is resolved by aligning further runs; Fig. 9 gives another example where an ambiguous local alignment is not resolved 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 positions of insertions that lengthen runs (insertions of the same symbol as in the run) and deletions within the runs where they occur.

For a length-$l_{\hat{\mathbf{Y}}}$ $\hat{\mathbf{Y}}$-run, its possible parent run/runs are categorized into the following cases, as shown in Fig. 7 (in all cases we give examples corresponding to the length-$l_{\hat{\mathbf{Y}}}$ $\hat{\mathbf{Y}}$-run being 00000):

- **Case 1:** The parent run is a "single run" with length $l_{\bar{\mathbf{X}}}$.
  - **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: $0000\cancel{0}0 \to 00000$.

[9] 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. Note that insertions that occur between two runs are therefore counted once in both $\bar{\mathbf{X}}$-runs, since they are in the extended run of each $\bar{\mathbf{X}}$-run.

														length
$\bar{\mathbf{X}}$	0	0	0	1	1	1	1	2	2	3	2	3	3	$n = 13$
edits	0	$0^{\downarrow 1}$	0	1	$\cancel{1}^{\downarrow 4}$	$\cancel{1}$	1	$\cancel{2}$	2	3	2	$3^{\downarrow 3}$	3	
$\bar{\mathbf{E}}$	η	$\eta\ \iota_1$	η	η	$\Delta\ \iota_4$	Δ	η	Δ	η	η	η	$\eta\ \iota_3$	η	$n + K_I = 16$
$\bar{\mathbf{Y}}$	0	0 1	0	1	4		1		2	3	2	3 3	3	$n - K_D + K_I = 13$
$\hat{\mathbf{E}}^C$	$-$	$-$	$-$	$-$	$\underline{\Delta}\ \underline{\iota_4}$	$\underline{\Delta}$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$n + K_I - \hat{K_I} = 14$
$\hat{\mathbf{E}}$	η	$\eta\ \iota_1$	η	η	η	η	η	Δ	η	η	η	$\eta\ \iota_3$	η	$n + \hat{K_I} = 15$
$\hat{\mathbf{Y}}$	0	0 1	0	1	1	1	1		2	3	2	3 3	3	$n - \hat{K_D} + \hat{K_I} = 14$

Fig. 5: Example of the defined file and edit sequences: 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 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 $\bar{C}^{K_I}$. 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 $K_I = 3$ insertions in the edit sequence, the length of the edit sequence $\bar{\mathbf{E}}$ equals $n + 3 = 16$. The resulting PosESS sequence $\bar{\mathbf{Y}}$ 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 the edit process 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 PreESS $\hat{\mathbf{Y}}$ resulting from operating $\hat{\mathbf{E}}$ on $\bar{\mathbf{X}}$.

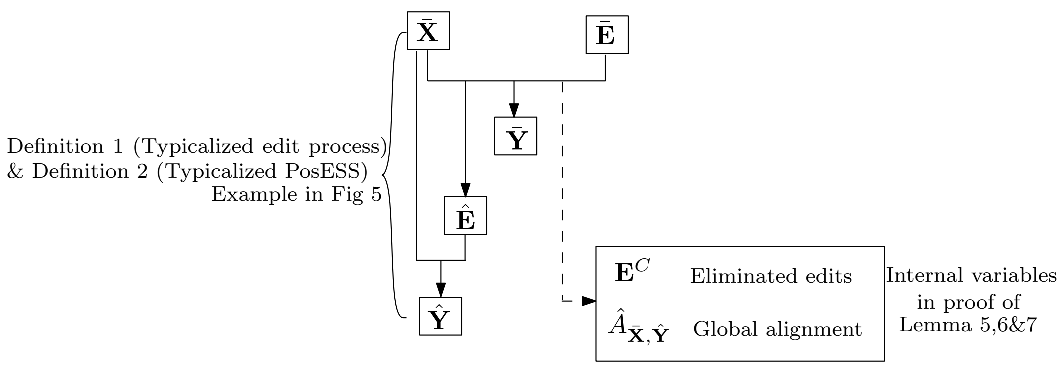

Fig. 6: The dependency of all the sequences and internal random variables for the proofs.

- **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 in the middle of a parent run breaks it into two runs. In this case, $l_{\bar{\mathbf{X}}} > l_{\hat{\mathbf{Y}}}$. Eg: $00000^{\downarrow 1}000 \to 000001000$. Moreover, the next run in $\hat{\mathbf{Y}}$ after this length-$l_{\hat{\mathbf{Y}}}$ $\hat{\mathbf{Y}}$-run is also aligned to this $\bar{\mathbf{X}}$-run.
- **Case 3 (multi-parent):** There are $2t+1$ parent $\bar{\mathbf{X}}$-runs of this $\hat{\mathbf{Y}}$-run. Of these parent $\bar{\mathbf{X}}$-runs, $t+1$ runs (the odd-numbered ones among the $2t + 1$ $\bar{\mathbf{X}}$-runs) comprise of the same symbol (0, in this example) as the corresponding $\hat{\mathbf{Y}}$-run, and are of lengths $l_1, \ldots, l_{t+1}$ respectively (say). Interleaved among these are the even-numbered $\bar{\mathbf{X}}$-runs, comprising of just one symbol each, that must be different from the symbols (0 in our example) 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 this $\hat{\mathbf{Y}}$-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: $00\cancel{1}00\cancel{2}0 \to 00000$.

Noting the parent run/runs 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, the runs in $\hat{\mathbf{Y}}$ can be "almost" aligned to the parent run/runs in $\bar{\mathbf{X}}$, except for the following two *ambiguous local alignment* events. We show later that with the help of some "small amount" additional information $H(A_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})$, $(\bar{\mathbf{X}}, \hat{\mathbf{Y}})$ can be aligned.

- Ambiguous local alignment type-1 Γ^1 $(l_{\bar{\mathbf{X}}} = l_{\hat{\mathbf{Y}}} - 1)$: Recall 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 $\bar{\mathbf{X}}$-run is the same as in Case 1.2 ($l_{\bar{\mathbf{X}}} = l_{\hat{\mathbf{Y}}} - 1$). Hence, when finding the length of the to-be-aligned $\bar{\mathbf{X}}$-run for a length-$l_{\hat{\mathbf{Y}}}$ $\hat{\mathbf{Y}}$-run to be $l_{\hat{\mathbf{Y}}} - 1$, 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 Case 2 ($l_{\bar{\mathbf{X}}} > l_{\hat{\mathbf{Y}}}$), when $l_{\bar{\mathbf{X}}} = l_{\hat{\mathbf{Y}}} + 1$ and the insertion of a different symbol occurs in front of the last symbol of the $\bar{\mathbf{X}}$-run, leading to a length-l $\hat{\mathbf{Y}}$-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, when finding the length of the to-be-aligned $\bar{\mathbf{X}}$-run for a length-$l_{\hat{\mathbf{Y}}}$ $\hat{\mathbf{Y}}$-run to be $l_{\hat{\mathbf{Y}}} + 1$, one can't tell immediately whether it is Case 1.3 or Case 2.

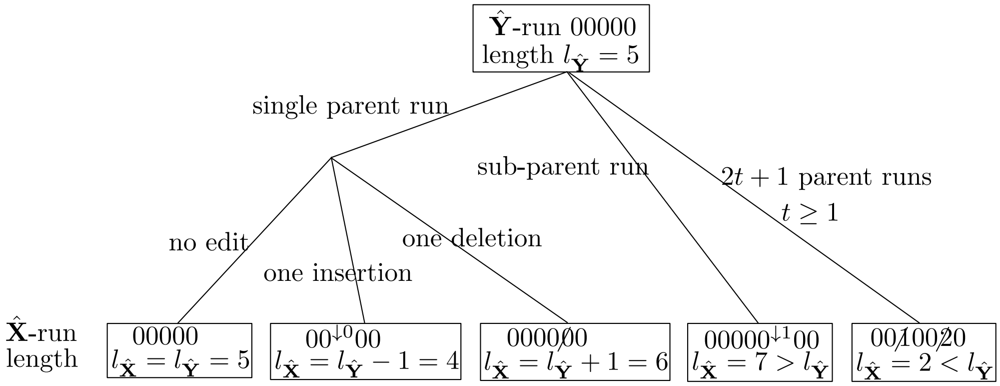

Fig. 7: Given a $\hat{\mathbf{Y}}$-run (00000) with length $l_{\hat{\mathbf{Y}}}$, its parent run may be a single run, a sub-run, or several runs. Because there can be no more than one edit in an extended run in the typicalized edit process, we can explicitly find the forms of the edits in different cases. If the parent run is a single run with length $l_{\bar{\mathbf{X}}}$, there may be no edit ($l_{\bar{\mathbf{X}}} = l_{\hat{\mathbf{Y}}}$); one insertion ($l_{\bar{\mathbf{X}}} = l_{\hat{\mathbf{Y}}} - 1$); or one deletion ($l_{\bar{\mathbf{X}}} = l_{\hat{\mathbf{Y}}} + 1$). If the parent run is a sub-run with length $l_{\bar{\mathbf{X}}}$, there must be one and only one insertion in the parent run, which breaks the parent run into two runs with length $l_{\hat{\mathbf{Y}}}$ and $l_{\bar{\mathbf{X}}} - l_{\hat{\mathbf{Y}}}$. In this case, $l_{\bar{\mathbf{X}}} > l_{\hat{\mathbf{Y}}}$. If the parent runs are several runs where the length of the first run is $l_{\bar{\mathbf{X}}}$, there must be $2t+1$ parent runs ($t \geq 1$), where the odd-number runs are runs with symbols the same as the $\hat{\mathbf{Y}}$-run, and the even-number runs are lenth-1 runs of symbols different from the $\hat{\mathbf{Y}}$-run. In this case, $l_{\bar{\mathbf{X}}} < l_{\hat{\mathbf{Y}}}$.

$\bar{\mathbf{X}}$	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:	$\cancel{0}$	0	0	$1^{\downarrow 0}$	1	1	1	2	$\cdots$					
Alignment 2:	0	$0^{\downarrow 1}$	0	1	1	1	1	2	$\cdots$					

Fig. 8: **Ambiguity resolved**: 1)There is an ambiguous local alignment type-2 event ($l_{\bar{\mathbf{X}}} = l_{\hat{\mathbf{Y}}} + 1$) 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 are comprised of 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 $\hat{\mathbf{Y}}$. 2)In fact, even if we examine the next one or two symbols in $\bar{\mathbf{X}}$ and $\hat{\mathbf{Y}}$, the local ambiguity is not resolved. 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) is comprised of, which means Case 2 (single-insertion breaking the $\bar{\mathbf{X}}$-run) is possible. 3)Ambiguity is resolved by 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 above. Since the second $\bar{\mathbf{X}}$-run had four 1's, the resulting $\hat{\mathbf{Y}}$-run have 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.

Note that 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. The example in Fig. 8 and Fig. 9 show both the scenario when an ambiguous local alignment is resolved later, and the scenario when an ambiguous local alignment leads to different global alignments.

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

$$\hat{\mathbf{Y}} = \hat{Y}(1)\hat{Y}(2)\ldots\hat{Y}(\rho_{\hat{\mathbf{Y}}}). \tag{5}$$

We then divide $\bar{\mathbf{X}}$ into "segments that leads to corresponding $\hat{\mathbf{Y}}$-runs" as

$$\bar{\mathbf{X}} = \bar{X}_{\hat{\mathbf{Y}}}(1)\bar{X}_{\hat{\mathbf{Y}}}(2)\ldots\bar{X}_{\hat{\mathbf{Y}}}(\rho_{\hat{\mathbf{Y}}}). \tag{6}$$

Note that $\bar{X}_{\hat{\mathbf{Y}}}(i)$'s are in general not runs of $\bar{\mathbf{X}}$. 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 it to join the segment of its right neighbouring run. The alignment of $\bar{\mathbf{X}}$ and $\hat{\mathbf{Y}}$ is defined by

$\bar{\mathbf{X}}$	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:	0	$0^{\downarrow 1}$	0	1	1	1	1	$\cancel{2}$	2	3	2	$3^{\downarrow 3}$	3	
Alignment 2:								$2^{\downarrow 3}$	2	3	$\cancel{2}$	3	3	

Fig. 9: **Ambiguity unresolved**: The edits in both Alignment 1 and alignment 2 convert $\bar{\mathbf{X}}$ to $\hat{\mathbf{Y}}$. The challenge therefore is to characterize the probability of such local ambiguity being globally unresolvable. This is the thrust of Lemma 5.

the vector of the lengths of the segments $\bar{X}_{\hat{\mathbf{Y}}}(i)$*'s,*

$$\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}} = (|\bar{X}_{\hat{\mathbf{Y}}}(1)|, |\bar{X}_{\hat{\mathbf{Y}}}(2)|, \ldots, |\bar{X}_{\hat{\mathbf{Y}}}(\rho_{\hat{\mathbf{Y}}})|). \tag{7}$$

Definition 5 (Partial alignment)**.** *For the subsequence of a typicalized PosESS* $\hat{\mathbf{Y}}$ *consisting of the first* $i_{\hat{\mathbf{Y}}}$ *runs* $\hat{Y}(1)\hat{Y}(2)\ldots\hat{Y}(i_{\hat{\mathbf{Y}}})$ *where* $i_{\hat{\mathbf{Y}}} \leq \rho_{\hat{\mathbf{Y}}}$*, suppose the segments of* $\bar{\mathbf{X}}$ *that lead to the* $\hat{\mathbf{Y}}$*-runs are* $\bar{X}_{\hat{\mathbf{Y}}}(1)\bar{X}_{\hat{\mathbf{Y}}}(2)\ldots\bar{X}_{\hat{\mathbf{Y}}}(i_{\hat{\mathbf{Y}}})$*. The partial alignment of* $\bar{\mathbf{X}}$ *and* $\hat{\mathbf{Y}}$ *upto "depth"* $i_{\hat{\mathbf{Y}}}$ *is defined by the vector of the lengths of the segments* $\bar{X}_{\hat{\mathbf{Y}}}(i)$*'s,*

$$\hat{A}^{i_{\hat{\mathbf{Y}}}}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}} = (|\bar{X}_{\hat{\mathbf{Y}}}(1)|, |\bar{X}_{\hat{\mathbf{Y}}}(2)|, \ldots, |\bar{X}_{\hat{\mathbf{Y}}}(i_{\hat{\mathbf{Y}}})|). \tag{8}$$

Recall that "nature's secret" is the uncertainty of the edit pattern given PreESS and PosESS. We now bound the "nature's secret" of the typicalized edit pattern $H(\hat{\mathbf{E}}|\bar{\mathbf{X}}, \hat{\mathbf{Y}})$ 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 the 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. $\lim_{n\to\infty} \frac{1}{n} H(\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}) \leq \mathcal{O}(\max(\epsilon, \delta)^2)$.

Proof: The intuition that the uncertainty $H(\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})$ of the global alignment is "small" is as follows. In any ambiguous local alignment event $\Gamma = \Gamma^1 \cup \Gamma^2$, one of the two edit patterns has an insertion and the other has a deletion. Hence "locally" the positions of the output $\hat{\mathbf{Y}}$ by applying these two edit patterns to $\bar{\mathbf{X}}$ differ by a shift of two positions. If the matching procedure described above in Fig. 10 keeps aligning $\bar{\mathbf{X}}$ w.r.t. $\hat{\mathbf{Y}}$ via both edit patterns, the ambiguity is still not resolved. That means we can find at least two distinct typicalized edit sequences that convert two "similar" sections of $\bar{\mathbf{X}}$ which differ by a shift of two positions 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 within a short block. This is because of the property of typicalized edits that "not too many" insertions or deletions (no contiguous insertions/deletions) can happen in a short block. Hence averaging over $\bar{\mathbf{X}}$, the probability that we need extra information to resolve ambiguous local alignments is "small".

In the following, we bound $H(\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})$ from above carefully. We first convert the uncertainty $H(A_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})$ averaging over PreESS $\bar{\mathbf{X}}$ and typicalized PosESS $\hat{\mathbf{Y}}$, to the number of "splits" (ambiguous local alignments unresolved) averaging over the PreESS $\bar{\mathbf{X}}$ and edit pattern $\bar{\mathbf{E}}$, as shown in Equation(9)–(14). Denote the number of $\bar{\mathbf{x}}$-runs by $\rho_{\bar{\mathbf{x}}}$. For $i = 1, 2, \ldots, \rho_{\bar{\mathbf{x}}}$, define the event $Ei(\bar{\mathbf{x}}, \bar{e})$ from the matching algorithm – after typicalizing $\bar{e}$ to $\hat{e}$ and processing $\hat{e}$ on $\bar{\mathbf{x}}$, the ith $\bar{\mathbf{x}}$-run encounter an ambiguous local alignment, and for the subsequence starting from the first symbol after the ith run and ending at the symbol before the next edit in $\hat{e}$ (we call the length of this block in $\bar{\mathbf{x}}$ the "gap"), the ambiguous edit pattern at the ith run can obtain the same $\hat{\mathbf{y}}$ through some typical edits. If $Ei(\bar{\mathbf{x}}, \bar{e})$ does occur, it may cause a split on the path of alignment where $\bar{e}$ belongs to, in which case one bit is needed to distinguish between the two ambiguous edit pattern. Hence, the total number of bits needed to distinguish the path/alignment associate with $\bar{e}$ from other paths splitting from it is bounded from above by $\sum_{i=1}^{\rho_{\bar{\mathbf{x}}}} \mathbb{1}_{Ei(\bar{\mathbf{x}},\bar{e})}$. For $i = 1, 2, \ldots, \rho_{\bar{\mathbf{x}}}$, denote the length of the ith $\bar{\mathbf{x}}$ -run by l_i. Conditioning on that an ambiguous local alignment $\Gamma_{(i)}$ occurs to the ith $\bar{\mathbf{x}}$-run, and the "gap" g from the symbol after the ith $\bar{\mathbf{x}}$-run until the symbol before the next edit, the probability $\Pr(Ei(\bar{\mathbf{x}}, \bar{e})|\Gamma_{(i)}, \text{longest gap } g)$ only depend on $\bar{\mathbf{x}}$ and g. We denote this probability averaged over $\bar{\mathbf{X}}$ by $\Pr_g = \sum_{\bar{\mathbf{x}}\in\bar{\mathbf{X}}} \Pr(\bar{\mathbf{x}}) \Pr(Ei(\bar{\mathbf{x}}, \bar{e})|\Gamma_{(i)}, \text{longest gap } g)$ and bound $\Pr_g$ later through some case analysis.

$$H(\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}) \overset{(a)}{=} \sum\nolimits_{\bar{\mathbf{x}}\in\bar{\mathbf{X}}} \Pr(\bar{\mathbf{x}}) \sum\nolimits_{\hat{\mathbf{y}}\in\hat{\mathbf{Y}}(\bar{\mathbf{x}})} \Pr(\hat{\mathbf{y}}|\bar{\mathbf{x}}) H(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}}) \tag{9}$$

$$\overset{(b)}{=} \sum\nolimits_{\bar{\mathbf{x}}\in\bar{\mathbf{X}}} \Pr(\bar{\mathbf{x}}) \sum\nolimits_{\hat{\mathbf{y}}\in\hat{\mathbf{Y}}(\bar{\mathbf{x}})} \left(\sum\nolimits_{\forall \bar{e}\in\bar{\mathbf{E}}, (\bar{\mathbf{x}},\bar{e})\to\hat{e}\to\hat{\mathbf{y}}} \Pr(\bar{e}) \right) H(\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}}) \tag{10}$$

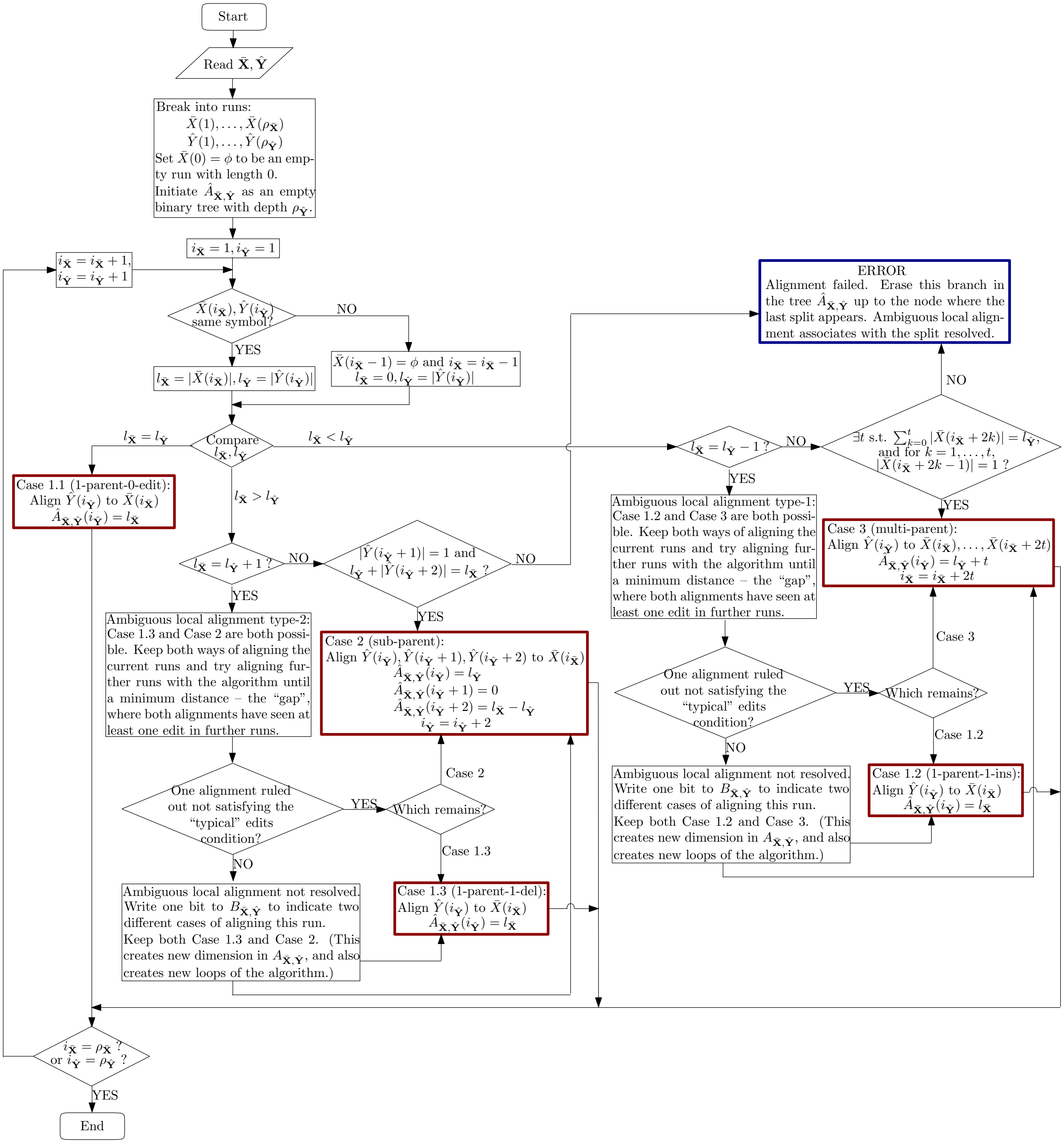

Fig. 10: The flowchart of the *align module* to align $\bar{\mathbf{X}}$ and $\hat{\mathbf{Y}}$: The module takes in $\bar{\mathbf{X}}$ and $\hat{\mathbf{Y}}$ as inputs, and outputs all the possible alignments $\hat{A}_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}$ as a binary tree of depth $\rho_{\hat{\mathbf{Y}}}$. 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 4; 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 5. 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.

$$\stackrel{(c)}{\leq} \sum_{\bar{\mathbf{x}}\in\bar{\mathbf{X}}} \Pr(\bar{\mathbf{x}}) \sum_{\hat{\mathbf{y}}\in\hat{\mathbf{Y}}(\bar{\mathbf{x}})} \left(\sum_{\forall \bar{e}\in\bar{\mathbf{E}},(\bar{\mathbf{x}},\bar{e})\to\hat{e}\to\hat{\mathbf{y}}} \Pr(\bar{e})\right) \sum_{\forall \text{ path } P_{\hat{A}}\in\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}}} \frac{\sum_{\forall\hat{e}\in P_{\hat{A}}}\sum_{\forall\bar{e},(\bar{\mathbf{x}},\bar{e})\to\hat{e}}\Pr(\bar{e})}{\sum_{\forall\bar{e}\in\bar{\mathbf{E}},(\bar{\mathbf{x}},\bar{e})\to\hat{e}\to\hat{\mathbf{y}}}\Pr(\bar{e})}\cdot N_{\text{split}}(P_{\hat{A}}) \tag{11}$$

$$\overset{(d)}{=} \sum_{\bar{\mathbf{x}}\in\bar{\mathbf{X}}} \Pr(\bar{\mathbf{x}}) \sum_{\hat{\mathbf{y}}\in\hat{\mathbf{Y}}(\bar{\mathbf{x}})} \sum_{\forall \text{ path } P_{\hat{A}}\in\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}}} \left(\sum_{\forall\hat{e}\in P_{\hat{A}}} \sum_{\forall\bar{e},(\bar{\mathbf{x}},\bar{e})\to\hat{e}} \Pr(\bar{e})\right) \cdot N_{\text{split}}(P_{\hat{A}}) \tag{12}$$

$$\overset{(e)}{=} \sum_{\bar{\mathbf{x}}\in\bar{\mathbf{X}}} \Pr(\bar{\mathbf{x}}) \sum_{\hat{\mathbf{y}}\in\hat{\mathbf{Y}}(\bar{\mathbf{x}})} \sum_{\forall \text{ path } P_{\hat{A}}\in\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}}} \sum_{\forall\hat{e}\in P_{\hat{A}}} \sum_{\forall\bar{e},(\bar{\mathbf{x}},\bar{e})\to\hat{e}} \left(\Pr(\bar{e}) \cdot N_{\text{split}}(P_{\hat{A}}(\bar{\mathbf{x}},\bar{e}))\right) \tag{13}$$

$$\overset{(f)}{=} \sum_{\bar{\mathbf{x}}\in\bar{\mathbf{X}}} \Pr(\bar{\mathbf{x}}) \sum_{\bar{e}\in\bar{\mathbf{E}}} \Pr(\bar{e}) \cdot N_{\text{split}}(P_{\hat{A}}(\bar{\mathbf{x}},\bar{e})) \tag{14}$$

$$\overset{(g)}{\leq} \sum_{\bar{\mathbf{x}}\in\bar{\mathbf{X}}} \Pr(\bar{\mathbf{x}}) \sum_{\bar{e}\in\bar{\mathbf{E}}} \Pr(\bar{e}) \sum_{i=1}^{\rho_{\bar{\mathbf{x}}}} \mathbb{1}_{Ei(\bar{\mathbf{x}},\bar{e})} \tag{15}$$

$$\overset{(h)}{=} \sum_{\bar{\mathbf{x}}\in\bar{\mathbf{X}}} \Pr(\bar{\mathbf{x}}) \sum_{\bar{e}\in\bar{\mathbf{E}}} \Pr(\bar{e}) \sum_{i=1}^{\rho_{\bar{\mathbf{x}}}} \Pr(Ei(\bar{\mathbf{x}},\bar{e})) \tag{16}$$

$$\overset{(i)}{=} \sum_{\bar{\mathbf{x}}\in\bar{\mathbf{X}}} \Pr(\bar{\mathbf{x}}) \sum_{\bar{e}\in\bar{\mathbf{E}}} \Pr(\bar{e}) \sum_{i=1}^{\rho_{\bar{\mathbf{x}}}} \sum_{g=1}^{\infty} \Pr(Ei(\bar{\mathbf{x}},\bar{e})|\Gamma_{(i)}, \text{longest gap } g) \Pr(\Gamma_{(i)}, \text{longest gap } g) \tag{17}$$

$$\overset{(j)}{=} \sum_{\bar{\mathbf{x}}\in\bar{\mathbf{X}}} \Pr(\bar{\mathbf{x}}) \sum_{i=1}^{\rho_{\bar{\mathbf{x}}}} \sum_{g=1}^{\infty} \Pr(Ei(\bar{\mathbf{x}},\bar{e})|\Gamma_{(i)}, \text{longest gap } g) \left(\sum_{\bar{e}\in\bar{\mathbf{E}}} \Pr(\bar{e}) \Pr(\Gamma_{(i)}, \text{longest gap } g)\right) \tag{18}$$

$$\overset{(k)}{\leq} \sum_{\bar{\mathbf{x}}\in\bar{\mathbf{X}}} \Pr(\bar{\mathbf{x}}) \sum_{i=1}^{\rho_{\bar{\mathbf{x}}}} \sum_{g=1}^{\infty} \Pr(Ei(\bar{\mathbf{x}},\bar{e})|\Gamma_{(i)}, \text{longest gap } g)(l_i+1)\max(\epsilon,\delta)^2 \tag{19}$$

$$\overset{(l)}{=} \max(\epsilon,\delta)^2 \sum_{i=1}^{\rho_{\bar{\mathbf{x}}}} (l_i+1) \sum_{g=1}^{\infty} \sum_{\bar{\mathbf{x}}\in\bar{\mathbf{X}}} \Pr(\bar{\mathbf{x}}) \Pr(Ei(\bar{\mathbf{x}},\bar{e})|\Gamma_{(i)}, \text{longest gap } g) \tag{20}$$

$$\overset{(m)}{\leq} \max(\epsilon,\delta)^2 2n \sum_{g=1}^{\infty} \sum_{\bar{\mathbf{x}}\in\bar{\mathbf{X}}} \Pr(\bar{\mathbf{x}}) \Pr(Ei(\bar{\mathbf{x}},\bar{e})|\Gamma_{(i)}, \text{longest gap } g) \tag{21}$$

$$= \max(\epsilon,\delta)^2 2n \sum_{g=1}^{\infty} \Pr_g. \tag{22}$$

In equality (a), the set $\hat{\mathbf{Y}}(\bar{\mathbf{x}})$ is obtained through typicalizing the set $\bar{\mathbf{Y}}(\bar{\mathbf{x}})$ – all the sequences $\bar{y}(\bar{\mathbf{x}})$ that resulting from processing any edit pattern $\bar{\mathbf{E}}$ on $\bar{\mathbf{x}}$. In equality (b), we replace $\Pr(\hat{\mathbf{y}}|\bar{\mathbf{x}})$ with the sum of the probabilities of all the edit patterns such that after typicalizing with $\bar{\mathbf{x}}$ and processing on $\bar{\mathbf{x}}$ obtains $\hat{\mathbf{y}}$. The inequality (c) follows by bounding the entropy of the tree $\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}}$ from above by the average of the number of splits $N_{\text{split}}(P_{\hat{A}})$ on all the paths. Note that a path of the tree $\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}}$ is a certain global alignment of $(\bar{\mathbf{x}},\hat{\mathbf{y}})$ – consisting of many typicalized edit pattern $\hat{e}$, the probability of which is the sum of the probabilities of all the $\bar{e}$ resulting in $\hat{e}$ after typicalizing. The equality (d) follows by directly canceling $\sum_{\forall\bar{e}\in\bar{\mathbf{E}},(\bar{\mathbf{x}},\bar{e})\to\hat{e}\to\hat{\mathbf{y}}} \Pr(\bar{e})$. Equality (e) and (f) follows because by fixing $\bar{\mathbf{x}}$ and $\bar{e}$, we fix a path on the tree $\hat{A}_{\bar{\mathbf{x}},\hat{\mathbf{y}}}$. Moreover, for all the $\bar{e}$'s which fixing on the same path, $N_{\text{split}}(P_{\hat{A}}(\bar{\mathbf{x}},\bar{e}))$'s equal.

In the following, we calculate $\Pr_g$ – conditioning on the occurrence of an ambiguous local alignment, the probability that the ambiguity is not resolved by continuing the matching process until the gap g – by breaking into four cases based on the type of the ambiguous local alignment and which edit is the edit that actual happens. $\Pr_g$ is the probability that averaging over $\bar{\mathbf{X}}$ and $\hat{\mathbf{E}}$, the path on the tree $A_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}$ splits into two branches at a node.

- **Ambiguous local alignment** 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 $\Gamma^1(\bar{\iota})$ – this possibility corresponds to an edit pattern resulting in $0^{\downarrow 0}x_1\cdots\to 00x_1\ldots$ with an insertion of 0. 2) Case $\Gamma^1(\bar{\Delta})$ – the other possibility corresponds to the edit pattern in which case x_1 is deleted and 0 combines with x_2 resulting in 00 in the corresponding locations in $\hat{\mathbf{Y}}$ – $0\cancel{x_1}0x_3\cdots\to 00x_3\ldots$. In this case x_2 must equal 0. In other words, if x_2 is not 0, this edit pattern is impossible and the ambiguity is resolved. Averaging over $p(\bar{\mathbf{X}})$, this 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 $1-\left(\frac{|\mathcal{A}|-1}{|\mathcal{A}|}\right)^2 = \frac{2|\mathcal{A}|-1}{|\mathcal{A}|^2}$.
  - **Case** $\Gamma^1(\bar{\iota})$**:** The actual edit $\hat{\mathbf{E}}$ is a single insertion $\bar{\iota}$, and until the gap g there is no other edit:

$$0^{\downarrow 0}x_10x_3x_4x_5\ldots x_g\cdots\to 00x_10x_3x_4x_5\ldots x_g\ldots. \tag{23}$$

In this case, the smallest g is 1, we denote $g=2t-1$ or $2t$, where $t=1,2,\ldots$. The ambiguous edit is a deletion of x_1 and should also result in the same $\hat{\mathbf{Y}}$ through some typical edits:

$$0\cancel{x_1}0x_3x_4x_5\dots x_g\dots \to 00x_3x_4x_5\dots x_g\dots \xrightarrow{\text{some typical edits}} 00x_10x_3x_4x_5\dots x_g\dots. \tag{24}$$

The symbol x_1 can equal any symbol from the alphabet but 0, 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 $x_3x_4x_5\dots x_g\dots$, the first g symbols of the resulting sequence should be $10x_3x_4x_5\dots x_g$ – a shift rightwards of two positions. 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. (These $\bar{\mathbf{X}}$'s are the ones that have splits in the tree $A_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}$ along the paths with the $\hat{\mathbf{E}}$ we are considering now.)

We first argue that the shift rightwards of two positions can't 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 the insertion makes the shift by one position, all the symbols after the insertion are the same and no other edits can happen (the symbols form a run). For example $x_3^{\downarrow 0}x_4x_5\dots x_g\dots \to 10x_3x_4x_5\dots x_g$, 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 $1^{\downarrow 0}x_4x_5\dots x_g\dots \to 101x_4x_5\dots x_g$. Also, x_4 also has to equal 1, because for typicalized edit patterns, x_4 can not 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.

We prove an upper bound on $\Pr_g$ by induction. Recall that either x_3 or x_4 has to equal $x_1 = 1$. Hence for $g = 1$, $\Pr_1 = 1 - \left(\frac{|\mathcal{A}|-1}{|\mathcal{A}|}\right)^2 = \frac{2|\mathcal{A}|-1}{|\mathcal{A}|^2}$. Assume for odd number $g = 2t-1$ where $t = 1, 2, \dots$, the sequence $x_3x_4x_5\dots x_g\dots$ can be converted to the shift of it rightwards by two positions up to the gap g – $10x_3x_4x_5\dots x_g$. We look for what condition should hold for the shifted sequence to be able to match up to the gap $g+2 = 2t+1$. Because we argued in the last paragraph that the position (index) of the sequence won't shift rightwards by two before the gap, the segment of sequence that convert to $10x_3x_4x_5\dots x_g$ ends at index at least $g+1$. If the index is $g+1$ – $x_3x_4x_5\dots x_{g+1}$ converts to $10x_3x_4x_5\dots x_g$, from the last paragraph, to match two more symbols we have $x_{g+3} = x_{g+2} = x_{g+1}$ with probability $\frac{1}{|\mathcal{A}|^2}$. If the index is greater than $g+1$, for example $g+2$ – $x_3x_4x_5\dots x_{g+2}$ converts to $10x_3x_4x_5\dots x_g$, 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}. By conditioning on whether x_{g+1} and x_{g+2} equal, the probability is $\frac{1}{|\mathcal{A}|} \cdot \left(1 - \left(\frac{|\mathcal{A}|-1}{|\mathcal{A}|}\right)^2\right) + \frac{|\mathcal{A}|-1}{|\mathcal{A}|} \cdot \left(1 - \left(\frac{|\mathcal{A}|-2}{|\mathcal{A}|}\right)^2\right) = \frac{4|\mathcal{A}|^2-6|\mathcal{A}|+3}{|\mathcal{A}|^3} < 1$. Hence 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 $\Pr_g = \Pr_{2t}$ 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 $t = 1, 2, \dots$.

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

$$0\cancel{x_1}0x_3x_4x_5\dots x_g\dots \to 00x_3x_4x_5\dots x_g\dots. \tag{25}$$

In this case, x_3 can be deleted and the smallest g is 2. We denote $g = 2t$ or $2t+1$, where $t = 1, 2, \dots$. The ambiguous edit is 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:

$$0^{\downarrow 0}x_10x_3x_4x_5\dots x_g\dots \to 00x_10x_3x_4x_5\dots x_g\dots \xrightarrow{\text{some typical edits}} 00x_3x_4x_5\dots x_g\dots. \tag{26}$$

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 $10x_3x_4x_5\dots x_g\dots$, the first $g-2$ symbols of the resulting sequence should be $x_3x_4x_5\dots x_g$ – a shift leftwards of two positions.

With similar arguments as Case $\Gamma^1(\bar{\iota})$, the position/index of the sequence won't shift leftwards by two positions to match the index of $\hat{\mathbf{Y}}$ before the actual edit pattern has the next edit (before 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 $\Pr_g = \Pr_{2t+1}$ 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 $t = 1, 2, \dots$.

- **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 an edit pattern resulting in $0\cancel{0}x_1\dots \to 0x_1\dots$ with an deletion of 0 in the run. 2) Case $\Gamma^2(\bar{\iota})$ – the other possibility corresponds to the edit pattern with an insertion of an symbol other than 0 in front of the last 0 in the run, breaking the $\bar{\mathbf{X}}$-run into two runs of 0 with length-$l_{\bar{\mathbf{X}}} - 1$ and length-1 – $0^{\downarrow \bar{\iota}}0x_1\dots \to 0\bar{\iota}0x_1\dots$.

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

$$0\not{0}x_1x_2x_3\dots x_g\dots \to 0x_1x_2x_3\dots x_g\dots. \tag{27}$$

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

$$0^{\downarrow x_1}0x_1x_2x_3\dots x_g\dots \to 0x_10x_1x_2x_3\dots x_g\dots \xrightarrow{\text{some typical edits}} 0x_1x_2x_3\dots x_g\dots. \tag{28}$$

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 $01x_2x_3x_4\dots x_g\dots$, the first $g-1$ symbols of the resulting sequence should be $x_2x_3x_4\dots x_g$ – a shift leftwards of two positions.

This is similar as Case $\Gamma^1(\bar{\Delta})$ – shift forwards of two positions. (The only difference here is the length of sequence needed to match after the shift is $g-1$ istead of $g-2$ in this case.) 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 $t = 1, 2, \dots$.

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

$$0^{\downarrow\bar{\iota}}0x_1x_2x_3\dots x_g\dots \to 0\bar{\iota}0x_1x_2x_3\dots x_g\dots. \tag{29}$$

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

$$0\not{0}x_1x_2x_3\dots x_g\dots \to 0x_1x_2x_3\dots x_g\dots \xrightarrow{\text{some typical edits}} 0\bar{\iota}0x_1x_2x_3\dots x_g\dots. \tag{30}$$

The ambiguity only exists if the inserted symbol $\bar{\iota}$ equals x_1. W.l.o.g., assume $\bar{\iota} = x_1 = 1$. From the above, there should be some typical edits such that after applying these edits to the sequence $x_2x_3\dots x_g\dots$, and the first $g+1$ symbols of the resulting sequence should be $01x_2x_3\dots x_g$ – a shift rightwards of two positions.

This is similar as Case $\Gamma^1(\bar{\iota})$ – shift rightwards of two positions. (The only difference here is the length of sequence needed to match after the shift is $g+1$ istead of g in this case.) 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 $t = 1, 2, \dots$.

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 $t = 1, 2, \dots$. Hence $H(A_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}) \le \max(\epsilon,\delta)^2 \cdot 2n \cdot \sum_{g=1}^{\infty} \Pr_g = \mathcal{O}(\max(\epsilon,\delta)^2)n$. □

Lemma 6 below characterizes the "nature's secret" of the typicalized edit process as defined in Definition 1.

Lemma 6. *$\lim_{n\to\infty} \frac{1}{n} H(\hat{\mathbf{E}}|\bar{\mathbf{X}},\hat{\mathbf{Y}}) \le C_{|\mathcal{A}|}(\delta+\epsilon) + \mathcal{O}(\max(\epsilon,\delta)^2)$, 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 $(\bar{\mathbf{x}},\hat{\mathbf{y}})$, the uncertainty in the typicalized edit pattern only lies in the uncertainty of the locations of single-deletions and the single-insertions of the same symbol (as in the run) within the $\bar{\mathbf{x}}$-runs. From the definition of the typicalized edit pattern, an $\bar{\mathbf{x}}$-run undergoes at most one edit. Hence, we define the following notations describing the edits from the $\bar{\mathbf{x}}$-runs perspective, which will be useful in calculating $H(\hat{\mathbf{E}}|\bar{\mathbf{X}},\bar{\mathbf{Y}},A_{\bar{\mathbf{X}},\bar{\mathbf{Y}}})$.

For any PreESS $\bar{\mathbf{x}}$, recall that we denote the number of runs in $\bar{\mathbf{x}}$ by $\rho_{\bar{\mathbf{x}}}$, and the run lengths by $\{l_1, l_2, \dots, l_{\rho_{\bar{\mathbf{x}}}}\}$. In the following, we derive the probability of insertions and deletions in the typicalized edit process from both *symbol-perspective* and *run-perspective*.

For the *symbol-perspective typicalized insertion/deletion probabilities*, for any $j = 1, 2, \dots, \rho_{\bar{\mathbf{x}}}$, denote $\hat{\delta}_j$ to be the probability that any specific symbol in the jth $\bar{\mathbf{x}}$-run is deleted, $\hat{\delta}_j = \delta(1-\epsilon-\delta)^{l_j+1} \in (\delta-(l_j+1)(\delta^2+\epsilon\delta), \delta)$. Similarly, denote $\hat{\epsilon}_j$ to be the probability that there is an insertion between two specific symbols in the extended run of the jth $\bar{\mathbf{x}}$-run, $\hat{\epsilon}_j = \epsilon(1-\epsilon-\delta)^{l_j+2} \in (\epsilon-(l_j+2)(\epsilon^2+\epsilon\delta), \epsilon)$. Actually, we only need $\hat{\delta}_j \le \delta$ and $\hat{\epsilon}_j \le \epsilon$ for upper bounding the "nature's secret". The specific distribution of the typicalized edit process is of interest for our future research on studying channel capacity of InDel channels.

Note that in the typicalized edit process, an $\bar{\mathbf{x}}$-run either undergoes a single-deletion or a single-insertion. Hence, we derive the insertion/deletion probabilities from the run-perspective. For any global alignment $a \in \{1, 2, \dots, \beta_{\bar{\mathbf{x}},\hat{\mathbf{y}}}\}$, denote $D_{(a)}^{\rho_{\bar{\mathbf{x}}}} \in \{0,1\}^{\rho_{\bar{\mathbf{x}}}}$ to be the *run-perspective single-deletion pattern*, where $D_{(a),j} = 1$ indicates there is one deletion in the jth $\bar{\mathbf{x}}$-run in global alignment a. Similarly, denote $I_{\text{same}(a)}^{\rho_{\bar{\mathbf{x}}}} \in \{0,1\}^{\rho_{\bar{\mathbf{x}}}}$ to be the *run-perspective single-same-symbol-insertion pattern*, where $I_{\text{same}(a),j} = 1$ indicates there is one insertion of the same symbol (insertion that lengthens the run) in the jth $\bar{\mathbf{x}}$-run in global alignment a. Dropping the subscript (a) in $D_{(a),j}$ and $I_{\text{same}(a),j}$, that is, D_j and $I_{\text{same},j}$ are indicating random variables of single-deletion and single-same-symbol-insertion in jth $\bar{\mathbf{x}}$-run averaging over all global alignments respectively. For a pair $(\bar{\mathbf{x}}, \hat{\mathbf{y}})$, denote the event that processing a typicalized edit pattern $\hat{\mathbf{e}}$ on $\bar{\mathbf{x}}$ leads to $\hat{\mathbf{y}}$, $p(\hat{\mathbf{y}}|\bar{\mathbf{x}}) = \sum_{\forall \hat{\mathbf{e}} \text{ s.t.}(\bar{\mathbf{x}},\hat{\mathbf{e}})\to\hat{\mathbf{y}}} p(\hat{\mathbf{e}})$. Moreover, all the typicalized edit patterns $\hat{\mathbf{e}}$ that processing $\bar{\mathbf{x}}$ to $\hat{\mathbf{y}}$ – $\{\forall \hat{\mathbf{e}} \text{ s.t.}(\bar{\mathbf{x}}, \hat{\mathbf{e}}) \to \hat{\mathbf{y}}\}$ – are classified into $\beta_{\bar{\mathbf{x}},\hat{\mathbf{y}}}$ groups $\{\hat{\mathbf{E}}_{(a)}\}$ based on the global alignments, where $\hat{\mathbf{E}}_{(a)}$ denotes the set of typicalized edit patterns $\hat{\mathbf{e}}$ that belongs to global alignment a of $(\bar{\mathbf{x}}, \hat{\mathbf{y}})$. Hence, for all $a \in \{1, 2, \ldots, \beta_{\bar{\mathbf{x}},\hat{\mathbf{y}}}\}$, $p(A_{\bar{\mathbf{x}},\hat{\mathbf{y}}} = a) = \left(\sum_{\forall \hat{\mathbf{e}}\in\hat{\mathbf{E}}_{(a)} \text{ s.t.}(\bar{\mathbf{x}},\hat{\mathbf{e}})\to\hat{\mathbf{y}}} p(\hat{\mathbf{e}})\right) / \left(\sum_{\forall \hat{\mathbf{e}} \text{ s.t.}(\bar{\mathbf{x}},\hat{\mathbf{e}})\to\hat{\mathbf{y}}} p(\hat{\mathbf{e}})\right) = \left(\sum_{\forall \hat{\mathbf{e}}\in\hat{\mathbf{E}}_{(a)} \text{ s.t.}(\bar{\mathbf{x}},\hat{\mathbf{e}})\to\hat{\mathbf{y}}} p(\hat{\mathbf{e}})\right) / p(\hat{\mathbf{y}}|\bar{\mathbf{x}})$. Hence, $\sum_{\hat{\mathbf{y}}} p(\hat{\mathbf{y}}|\bar{\mathbf{x}}) \sum_{a=1}^{\beta_{\bar{\mathbf{x}},\hat{\mathbf{y}}}} p(A_{\bar{\mathbf{x}},\hat{\mathbf{y}}} = a) p(D_{(a),j} = 1) = \sum_{\hat{\mathbf{y}}} \sum_{a=1}^{\beta_{\bar{\mathbf{x}},\hat{\mathbf{y}}}} \sum_{\forall \hat{\mathbf{e}}\in\hat{\mathbf{E}}_{(a)} \text{ s.t.}(\bar{\mathbf{x}},\hat{\mathbf{e}})\to\hat{\mathbf{y}}} p(\hat{\mathbf{e}}) p(D_{(a),j} = 1) = \sum_{\hat{\mathbf{e}}} p(\hat{\mathbf{e}}) p(D_j = 1)$ is the probability that there is one deletion in the jth $\bar{\mathbf{x}}$-run averaging over all the typicalized edit patterns, and equals $l_j \hat{\delta}_j$. Similarly, $\sum_{\hat{\mathbf{y}}} p(\hat{\mathbf{y}}|\bar{\mathbf{x}}) \sum_{a=1}^{\beta_{\bar{\mathbf{x}},\hat{\mathbf{y}}}} p(A_{\bar{\mathbf{x}},\hat{\mathbf{y}}} = a) p(I_{\text{same}(a),j} = 1) = \sum_{\hat{\mathbf{e}}} p(\hat{\mathbf{e}}) p(I_{\text{same},j} = 1)$ is the probability that there is an insertion of the same symbol in the jth $\bar{\mathbf{x}}$-run averaging over all the typicalized edit patterns, and equals $\frac{1}{|\mathcal{A}|}(l_j + 1)\hat{\epsilon}_j$.

$$
\begin{aligned}
H(\hat{\mathbf{E}}|\bar{\mathbf{X}}, \hat{\mathbf{Y}}, A_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}) &= \sum_{\bar{\mathbf{x}},\hat{\mathbf{y}},a} p(\bar{\mathbf{x}}, \hat{\mathbf{y}}, a) H(\hat{\mathbf{E}}|\bar{\mathbf{x}}, \hat{\mathbf{y}}, a) && (31)\\
&= \sum_{\bar{\mathbf{x}},\hat{\mathbf{y}},a} p(\bar{\mathbf{x}}, \hat{\mathbf{y}}) p(a|\bar{\mathbf{x}}, \hat{\mathbf{y}}) H(\hat{\mathbf{E}}|\bar{\mathbf{x}}, \hat{\mathbf{y}}, a) && (32)\\
&= \sum_{\bar{\mathbf{x}},\hat{\mathbf{y}}} p(\bar{\mathbf{x}}, \hat{\mathbf{y}}) \sum_{a=1}^{\beta_{\bar{\mathbf{x}},\hat{\mathbf{y}}}} p(A_{\bar{\mathbf{x}},\hat{\mathbf{y}}} = a) H(\hat{\mathbf{E}}|\bar{\mathbf{x}}, \hat{\mathbf{y}}, a) && (33)\\
&\overset{(a)}{=} \sum_{\bar{\mathbf{x}},\hat{\mathbf{y}}} p(\bar{\mathbf{x}}, \hat{\mathbf{y}}) \sum_{a=1}^{\beta_{\bar{\mathbf{x}},\hat{\mathbf{y}}}} p(A_{\bar{\mathbf{x}},\hat{\mathbf{y}}} = a) \sum_{j=1}^{\rho_{\bar{\mathbf{x}}}} \left(D_{(a),j} \log l_j + I_{\text{same}(a),j} \log (l_j + 1)\right) && (34)\\
&= \sum_{\bar{\mathbf{x}}} p(\bar{\mathbf{x}}) \sum_{\hat{\mathbf{y}}} p(\hat{\mathbf{y}}|\bar{\mathbf{x}}) \sum_{a=1}^{\beta_{\bar{\mathbf{x}},\hat{\mathbf{y}}}} p(A_{\bar{\mathbf{x}},\hat{\mathbf{y}}} = a) \sum_{j=1}^{\rho_{\bar{\mathbf{x}}}} \left(D_{(a),j} \log l_j + I_{\text{same}(a),j} \log (l_j + 1)\right) && (35)\\
&= \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_{a=1}^{\beta_{\bar{\mathbf{x}},\hat{\mathbf{y}}}} p(A_{\bar{\mathbf{x}},\hat{\mathbf{y}}} = a) \left(p(D_{(a),j} = 1) \log l_j + p(I_{\text{same}(a),j} = 1) \log (l_j + 1)\right) && (36)\\
&\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) && (37)\\
&\overset{(c)}{\leq} \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) && (38)\\
&\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 + \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) && (39)\\
&\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 && (40)
\end{aligned}
$$

where step (a) is because when the global alignment of $(\bar{\mathbf{x}}, \hat{\mathbf{y}})$ is known, the uncertainty only lies in the edit-positions in those $\bar{\mathbf{x}}$-runs undergoing single-deletion and single-same-symbol-insertion. Step (b) comes from the analysis in the last paragraph. Step (c) is because $\hat{\delta}_j \in (\delta - (l_{(j)} + 1)(\delta^2 + \epsilon\delta), \delta)$ and $\hat{\epsilon}_j \in (\epsilon - (l_{(j)} + 2)(\epsilon^2 + \epsilon\delta), \epsilon)$. (In fact, it is straightforward that $\hat{\delta}_j \leq \delta$ and $\hat{\epsilon}_j \leq \epsilon$, because the typicalized edit pattern is obtained from the original edit pattern through eliminating some edits.) Step (d) is because $\sum_{\bar{\mathbf{x}}} p(\bar{\mathbf{x}}) \sum_{j=1}^{\rho_{\bar{\mathbf{x}}}} l_j \log l_j = \sum_{l=1}^{\infty} \frac{np(l)}{E[L]} l \log l$, 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. Similarly 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) comes from changing the index $l + 1$ to l and some calculation.

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

□

In the following Lemma 7, we show that the nature's secret for the original edit process is "close" to the nature's secret of the typicalized edit process. We first reprise a useful fact from [21].

Fact 2. *[21][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 $U \overset{V}{\longleftrightarrow} \hat{U}$.) Then*

$$|H(U) - H(\hat{U})| \leq H(V). \tag{41}$$

Lemma 7. $\lim_{n\to\infty} \frac{1}{n}|H(\bar{\mathbf{E}}|\bar{\mathbf{X}},\bar{\mathbf{Y}}) - H(\hat{\mathbf{E}}|\bar{\mathbf{X}},\hat{\mathbf{Y}})| \leq 56\max(\epsilon,\delta)^{2-\tau} + \mathcal{O}(\max(\epsilon,\delta)^2)$ *for any* $\tau > 0$.

Proof: We use Fact 2 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 2, and further, show below that the conditions required in Fact 2 are satisfied. Similarly, by mapping $(\bar{\mathbf{X}},\bar{\mathbf{Y}})$ as U, $(\bar{\mathbf{X}},\hat{\mathbf{Y}})$ as $\hat{U}$, and $(\hat{\mathbf{E}}^C, A_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})$ as V in Fact 2, and showing below that the conditions required in Fact 2 are also satisfied, we can bound $|H(\bar{\mathbf{X}},\hat{\mathbf{Y}}) - H(\bar{\mathbf{X}},\bar{\mathbf{Y}})|$ by $H(\hat{\mathbf{E}}^C, A_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})$. Hence, $|H(\bar{\mathbf{E}}|\bar{\mathbf{X}},\bar{\mathbf{Y}}) - H(\hat{\mathbf{E}}|\bar{\mathbf{X}},\hat{\mathbf{Y}})| = |(H(\bar{\mathbf{E}},\bar{\mathbf{X}},\bar{\mathbf{Y}}) - H(\hat{\mathbf{E}},\bar{\mathbf{X}},\hat{\mathbf{Y}})) + (H(\bar{\mathbf{X}},\hat{\mathbf{Y}}) - H(\bar{\mathbf{X}},\bar{\mathbf{Y}}))| \leq H(\hat{\mathbf{E}}^C) + H(\hat{\mathbf{E}}^C, A_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}) \leq 2H(\hat{\mathbf{E}}^C) + H(A_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})$.

The detailed reasoning for the two pairs of the relations by the above mapping in Fact 2 is as follows.

- $(\bar{\mathbf{E}},\bar{\mathbf{X}},\bar{\mathbf{Y}}) \overset{\hat{\mathbf{E}}^C}{\longleftrightarrow} (\hat{\mathbf{E}},\bar{\mathbf{X}},\hat{\mathbf{Y}})$
  - "$\rightarrow$": The typicalized edit pattern $\hat{\mathbf{E}}$ as given in Definition 1 is a deterministic function of $\bar{\mathbf{E}}$ and $\bar{\mathbf{X}}$. Then given $\hat{\mathbf{E}}$ and $\bar{\mathbf{X}}$, one can compute the typicalized PosESS $\hat{\mathbf{Y}}$ as noted in Definition 2.
  - "$\leftarrow$": To show that $(\hat{\mathbf{E}},\bar{\mathbf{X}},\hat{\mathbf{Y}})$ is a deterministic function of $(\bar{\mathbf{E}},\bar{\mathbf{X}},\bar{\mathbf{Y}})$ and $\hat{\mathbf{E}}^C$, we proceed as follows. We firstly align the '$-$'s and '$\underline{\bar{\Delta}}$'s in $\hat{\mathbf{E}}^C$ with the '$\bar{\eta}$'s and the '$\bar{\Delta}$'s in $\hat{\mathbf{E}}$. We then obtain $\bar{\mathbf{E}}$ from $\hat{\mathbf{E}}$ by changing the '$\bar{\eta}$'s to '$\bar{\Delta}$'s where the corresponding symbol is $\underline{\bar{\Delta}}$s in $\hat{\mathbf{E}}^C$, and inserting insertion edits '$\bar{\iota}$'s of the corresponding content back where there are '$\underline{\bar{\iota}}$'s in $\hat{\mathbf{E}}^C$. The corresponding example is shown in Fig. 11. The intuition is that the original edit pattern $\bar{\mathbf{E}}$ is a "union" of the typicalized edits $\hat{\mathbf{E}}$ and the eliminated edits stored in the complement of the typicalized edit pattern $\hat{\mathbf{E}}^C$. After determining $\bar{\mathbf{E}}$, $\bar{\mathbf{Y}}$ can be determined from $(\bar{\mathbf{X}},\bar{\mathbf{E}})$.
- $(\bar{\mathbf{X}},\bar{\mathbf{Y}}) \xleftrightarrow{(\hat{\mathbf{E}}^C, A_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})} (\bar{\mathbf{X}},\hat{\mathbf{Y}})$
  - "$\leftarrow$": With $A_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}$, the $\hat{\mathbf{Y}}$-runs can be aligned to parent run/runs in $\bar{\mathbf{X}}$ without any ambiguity. Indeed, this is the content of Lemma 6. Also, the atypical edits $\hat{\mathbf{E}}^C$ can be aligned to $\bar{\mathbf{X}}$. Then given the typicalized PosESS $\hat{\mathbf{Y}}$ and the atypical edits $\hat{\mathbf{E}}^C$, one can reconstruct $\bar{\mathbf{Y}}$ as follows. If the corresponding sections in $\hat{\mathbf{E}}^C$ for a $\bar{\mathbf{X}}$-run-$\hat{\mathbf{Y}}$-run match is "empty" (comprises only of '$-$'), then we reconstruct the run/runs of $\bar{\mathbf{Y}}$ as the same as the run/runs in $\hat{\mathbf{Y}}$. For the sections where the atypical edits $\hat{\mathbf{E}}^C$ are nonempty (has some eliminated insertions '$\underline{\bar{\iota}}$'/deletions '$\underline{\bar{\Delta}}$'), the corresponding $\bar{\mathbf{X}}$ undergoes some atypical edits in $\bar{\mathbf{E}}$, which are all eliminated in $\hat{\mathbf{E}}$. Hence the corresponding $\hat{\mathbf{Y}}$-run is exactly the same as the $\bar{\mathbf{X}}$-run. To reconstruct these atypical runs in $\bar{\mathbf{Y}}$, we only need to apply the eliminated edits specified in $\hat{\mathbf{E}}^C$ back to the corresponding $\bar{\mathbf{X}}$-runs. The corresponding example is shown in Fig. 12.
  - "$\rightarrow$": Although $(\bar{\mathbf{X}},\bar{\mathbf{Y}})$ are in general hard to align, with the aid of $\hat{\mathbf{E}}^C$, the 0-subsequences of $\hat{\mathbf{E}}^C$ correspond to no edit-elimination parts in $\bar{\mathbf{X}}$. Hence the corresponding parts in $\bar{\mathbf{Y}}$ remain the same in $\hat{\mathbf{Y}}$. The nonzero entries in $\hat{\mathbf{E}}^C$ specify the specific edit pattern in the $\bar{\mathbf{X}}$-runs where there are edit-eliminations. Those $\bar{\mathbf{X}}$-runs undergo no edits in $\hat{\mathbf{Y}}$. The alignment $A_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}$ helps with alignment $\hat{\mathbf{E}}^C$ to the $\bar{\mathbf{X}}$-runs. The corresponding example is shown in Fig. 13.

eliminated insertions are restored by inserting $\bar{\iota}$ back

$$\begin{array}{cccccccccccccccccc}
\hat{\mathbf{E}}^C & - & - & & - & - & \underline{\bar{\Delta}} & \boxed{\underline{\bar{\iota}}_4} & \underline{\bar{\Delta}} & - & - & - & - & - & - & & - \\
\hat{\mathbf{E}} & \bar{\eta} & \bar{\eta} & \bar{\iota}_1 & \bar{\eta} & \bar{\eta} & \bar{\eta} & & \bar{\eta} & \bar{\eta} & \bar{\Delta} & \bar{\eta} & \bar{\eta} & \bar{\eta} & \bar{\eta} & \bar{\iota}_3 & \bar{\eta} \\
\bar{\mathbf{E}} & \bar{\eta} & \bar{\eta} & \bar{\iota}_1 & \bar{\eta} & \bar{\eta} & \bar{\Delta} & \boxed{\bar{\iota}_4} & \bar{\Delta} & \bar{\eta} & \bar{\Delta} & \bar{\eta} & \bar{\eta} & \bar{\eta} & \bar{\eta} & \bar{\iota}_3 & \bar{\eta}
\end{array}$$

eliminated deletions are restored by replacing $\bar{\eta}$ with $\bar{\Delta}$

Fig. 11: Example of $\bar{\mathbf{E}} \overset{\hat{\mathbf{E}}^C}{\longleftarrow} \hat{\mathbf{E}}$

In $\hat{\mathbf{E}}^C$, there is an elimination of a deletion with probability $\zeta_j^{\underline{\bar{\Delta}}} = \delta - \delta(1-\epsilon-\delta)^{l_{(j)}+1} \leq (l_{(j)}+1)(\epsilon\delta+\delta^2)$, where $l_{(j)}$ is the length of the run where $\bar{E}_j$ occurs. Averaging over $\bar{\mathbf{X}}$, denote the run length random variable by L, the probability that a deletion in $\bar{\mathbf{E}}$ is eliminated is $\zeta^{\underline{\bar{\Delta}}} = E_L[\zeta_j^{\underline{\bar{\Delta}}}] \leq (E[L]+1)(\epsilon\delta+\delta^2)$. Note that $E[L] = \frac{|\mathcal{A}|}{|\mathcal{A}|-1} \leq 2$, where equality holds when $|\mathcal{A}| = 2$. Hence $\zeta^{\underline{\bar{\Delta}}} \leq 3(\epsilon\delta+\delta^2) \leq 6\max(\epsilon,\delta)^2$

Similarly, there is an elimination of an insertion in $\hat{\mathbf{E}}^C$ with probability $\zeta_j^{\underline{\bar{\iota}}} = \epsilon - \epsilon(1-\epsilon-\delta)^{l_{(j)}+2} \leq (l_{(j)}+2)(\epsilon\delta+\epsilon^2)$, where $l_{(j)}$ is the length of the run where $\bar{E}_j$ occurs. Averaging over $\bar{\mathbf{X}}$, denote the run length random variable by L, the probability that an insertion in $\bar{\mathbf{E}}$ is eliminated is $\zeta^{\underline{\bar{\iota}}} = E_L[\zeta_j^{\underline{\bar{\iota}}}] \leq (E[L]+2)(\epsilon\delta+\epsilon^2)$. Hence $\zeta^{\underline{\bar{\iota}}} \leq 4(\epsilon\delta+\epsilon^2) \leq 8\max(\epsilon,\delta)^2$

$$\text{with } A_{\bar{\mathbf{X}},\hat{\mathbf{Y}}}\text{, the alignment of } \bar{\mathbf{X}},\hat{\mathbf{Y}} \text{ is known}$$

$$\begin{array}{c|cccc|ccccc|cc|c|c|ccc}
\bar{\mathbf{X}} & 0 & 0 & & 0 & 1 & 1 & & 1 & 1 & 2 & 2 & 3 & 2 & 3 & 3 & \\
\hat{\mathbf{Y}} & 0 & 0 & 1 & 0 & 1 & 1 & & 1 & 1 & 2 & & 3 & 2 & 3 & 3 & 3 \\
\hat{\mathbf{E}}^C & - & - & & - & - & \underline{\bar{\Delta}} & \underline{\bar{\iota}}_4 & \underline{\bar{\Delta}} & - & - & - & - & - & - & - & \\
\bar{\mathbf{Y}} & 0 & 0 & 1 & 0 & 1 & & 4 & & 1 & 2 & & 3 & 2 & 3 & 3 & 3
\end{array}$$

in the run-matches where $\hat{\mathbf{E}}^C$ are all $-$'s, $\bar{\mathbf{Y}}$ is the same as $\hat{\mathbf{Y}}$
otherwise, apply the eliminated edits specified in $\hat{\mathbf{E}}^C$ back to get $\bar{\mathbf{Y}}$

Fig. 12: Example of $\bar{\mathbf{Y}} \xleftarrow{(\hat{\mathbf{E}}^C, A_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})} \hat{\mathbf{Y}}$

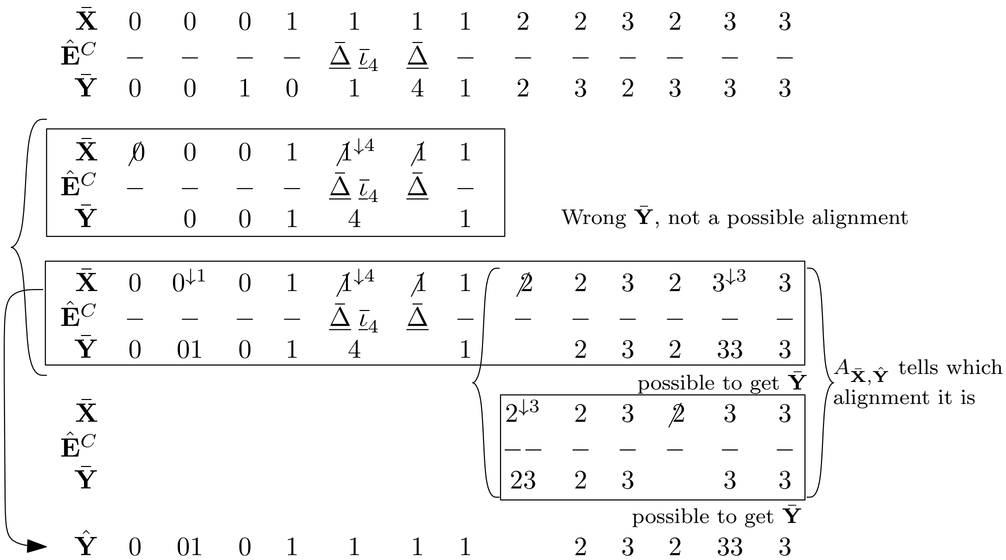

Fig. 13: Example of $\bar{\mathbf{Y}} \xrightarrow{(\hat{\mathbf{E}}^C, A_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})} \hat{\mathbf{Y}}$

Recall Definition 3 that $\hat{\mathbf{E}}^C = (\underline{\bar{O}}^{n+K_I-\hat{K}_I}, \underline{\bar{C}}^{K_I-\hat{K}_I})$. By similar calculation as Equation 4 in Lemma 4,

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

Hence,

$$\begin{aligned}
H(\hat{\mathbf{E}}^C) &= H(\underline{\bar{O}}^{n+K_I-\hat{K}_I}, \underline{\bar{C}}^{K_I-\hat{K}_I}) \\
&= H(\underline{\bar{O}}^{n+K_I-\hat{K}_I}) + H(\underline{\bar{C}}^{K_I-\hat{K}_I} | \underline{\bar{O}}^{n+K_I-\hat{K}_I}) \\
&\overset{(a)}{=} (n + E[K_I] - E[\hat{K}_I])H(\underline{\bar{O}}_1) + H(\underline{\bar{C}}^{K_I-\hat{K}_I} | (K_I - \hat{K}_I)) \\
&= (n + E[K_I] - E[\hat{K}_I])H(\underline{\bar{O}}_1) + \Big(E[K_I] - E[\hat{K}_I]\Big)\log|\mathcal{A}| \\
&\le \frac{n+\epsilon}{1-\epsilon}H(\underline{\bar{O}}_1) + \frac{n+\epsilon}{1-\epsilon}\zeta^{\underline{\bar{\iota}}}\log|\mathcal{A}|
\end{aligned}$$

$$\leq \frac{n+\epsilon}{1-\epsilon}\left(28\max{(\epsilon,\delta)}^{2-\tau}+\mathcal{O}(\max{(\epsilon,\delta)}^2)+8\max{(\epsilon,\delta)}^2\log|\mathcal{A}|\right) \quad = \frac{n+\epsilon}{1-\epsilon}\left(28\max{(\epsilon,\delta)}^{2-\tau}+\mathcal{O}(\max{(\epsilon,\delta)}^2)\right)$$

where step (a) is by Theorem 1.

Hence, $\lim_{n\to\infty}\frac{1}{n}|H(\bar{\mathbf{E}}|\bar{\mathbf{X}},\bar{\mathbf{Y}})-H(\hat{\mathbf{E}}|\bar{\mathbf{X}},\hat{\mathbf{Y}})|\leq \lim_{n\to\infty}\frac{1}{n}\left(2H(\hat{\mathbf{E}}^C)+H(A_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})\right)\leq 56\max{(\epsilon,\delta)}^{2-\tau}+\mathcal{O}(\max{(\epsilon,\delta)}^2)$ for any $\tau>0$. (Recall in the proof of Lemma 6 we've shown that $H(A_{\bar{\mathbf{X}},\hat{\mathbf{Y}}})\leq\mathcal{O}(\max{(\epsilon,\delta)}^2)n$.)

□

Remark: For our purpose of finding a lower bound on the achievable rate, we only need one direction, that is, $\lim_{n\to\infty}\frac{1}{n}(H(\bar{\mathbf{E}}|\bar{\mathbf{X}},\bar{\mathbf{Y}})-H(\hat{\mathbf{E}}|\bar{\mathbf{X}},\hat{\mathbf{Y}}))\geq -56\max{(\epsilon,\delta)}^{2-\tau}+\mathcal{O}(\max{(\epsilon,\delta)}^2)$. Lemma 7 gives a stronger statement and will be useful for our ongoing research on insertion-deletion channel capacity.

Theorem 8 below is the main theorem characterizing the information-theoretic lower bound of the optimal rate for RPES-LtRRID process.

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

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

$$\lim_{n\to\infty}\frac{1}{n}H(\bar{\mathbf{Y}}|\bar{\mathbf{X}})=\lim_{n\to\infty}\frac{1}{n}[H(\bar{\mathbf{E}}|\bar{\mathbf{X}})+H(\bar{\mathbf{Y}}|\bar{\mathbf{E}},\bar{\mathbf{X}})-H(\bar{\mathbf{E}}|\bar{\mathbf{X}},\bar{\mathbf{Y}})] \tag{42}$$

$$=\lim_{n\to\infty}\frac{1}{n}[H(\bar{\mathbf{E}})-H(\bar{\mathbf{E}}|\bar{\mathbf{X}},\bar{\mathbf{Y}})] \tag{43}$$

$$=\lim_{n\to\infty}\frac{1}{n}H(\bar{\mathbf{E}})-\lim_{n\to\infty}\frac{1}{n}H(\hat{\mathbf{E}}|\bar{\mathbf{X}},\hat{\mathbf{Y}})+\lim_{n\to\infty}\frac{1}{n}(H(\bar{\mathbf{E}}|\bar{\mathbf{X}},\bar{\mathbf{Y}})-H(\hat{\mathbf{E}}|\bar{\mathbf{X}},\hat{\mathbf{Y}})) \tag{44}$$

$$\geq H(\delta)+H(\epsilon)+\epsilon\log|\mathcal{A}|+2\min(\epsilon,\delta)^{2-\tau}-(\delta+\epsilon)C_{|\mathcal{A}|}-56\max{(\epsilon,\delta)}^{2-\tau}+\mathcal{O}(\max{(\epsilon,\delta)}^2) \tag{45}$$

$$\geq H(\delta)+H(\epsilon)+\epsilon\log|\mathcal{A}|-(\delta+\epsilon)C_{|\mathcal{A}|}-56\max{(\epsilon,\delta)}^{2-\tau}+\mathcal{O}(\max{(\epsilon,\delta)}^2) \tag{46}$$

□

Remark: When $\epsilon=0$ and $|\mathcal{A}|=2$, our result matches with result in Corollary IV.5. for the binary deletion channel in [16].

B. APES-AID Process

Given an arbitrary pre-edit source sequence $\mathbf{X}\in\mathcal{A}^n$, recall that the $\mathbf{X}$*-post-edit set* $\mathcal{Y}_{\epsilon,\delta}(\mathbf{X})$ denotes the set of all sequences over $\mathcal{A}$ that may be obtained from $\mathbf{X}$ via an arbitrary (ϵ,δ)-InDel process. For zero-error decodability, The encoder needs to send $\log|\mathcal{Y}_{\epsilon,\delta}(\mathbf{X})|$ bits to decoder. The larger the $\mathbf{X}$-post-edit set, 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 $\mathbf{X}$-post-edit set.

In two special cases of the edit process, the arbitrary ϵ-insertion process and the arbitrary δ-deletion process, the sizes of the post-edit sets have been well studied in literature. We here present the results in [19], [20] using our notation. For the arbitrary ϵ-insertion process, the size of the post-edit set $|\mathcal{Y}_{\epsilon,0}(\mathbf{X})|=\sum_{j=0}^{\epsilon n}\binom{n+\epsilon n}{j}(|\mathcal{A}|-1)^j\geq\binom{n+\epsilon n}{\epsilon n}(|\mathcal{A}|-1)^{\epsilon n}$ 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 the two special cases.

For an arbitrary ϵ-insertion process, consider a PreESS that we denote $\mathbf{X}_\alpha$, which is a single length-n run of the same symbol $\alpha\in\mathcal{A}$. Consider insertions of the form that 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 $\mathbf{Y}$ are all distinct. The number of such insertion patterns is $\binom{n+\epsilon n}{\epsilon n}(|\mathcal{A}|-1)^{\epsilon n}$. 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 – $\frac{1}{n}\log|\mathcal{Y}_{\epsilon,0}(\mathbf{X}_\alpha)|$, is asymptotically $(1+\epsilon)H(\frac{\epsilon}{1+\epsilon})+\epsilon\log(|\mathcal{A}|-1)$ by Stirling's approximation [24].

For an arbitrary δ-deletion processes, consider a PreESS that we denoted $\mathbf{X}_{\text{diff}}$, where each symbol is different from the preceding one, i.e., $\mathbf{X}_{\text{diff}}$ consists of n length-1 runs. Consider the set of deletion patterns which delet an arbitrary subset of δn non-pairwise-contiguous symbols from $\mathbf{X}_{\text{diff}}$. Note that each such deletion pattern results in a distinct PosESS $\mathbf{Y}$. The number of these deletion patterns is $\binom{n-\delta n}{\delta n}$. The corresponding lower bound on the optimal achievable rate – $\frac{1}{n}\log|\mathcal{Y}_{0,\delta}(\mathbf{X}_{\text{diff}})|$, is asymptotically $(1-\delta)H(\frac{\delta}{1-\delta})$ by Stirling's approximation [24].

To our best knowledge, there is no literature on the bounds for the scenario with both insertions and deletions. In the Theorem 9 below, we derive a lower bound on the achievable rate, by constructing a PreESS $\mathbf{X}_{\text{LB}}$ and a subset of InDel patterns, such that any of the InDel patterns in the subset, applied to $\mathbf{X}_{\text{LB}}$, results in a distinct PosESS $\mathbf{Y}$.

Theorem 9. *The optimal transmission rate of APES-AID process* $R^*_{\epsilon,\delta} \geq H(\delta)+H(\epsilon)+\epsilon\log|\mathcal{A}|-\frac{2}{|\mathcal{A}|}\epsilon-(2\log e)\max(\epsilon,\delta)^2+\mathcal{O}(\max(\epsilon,\delta)^3)+\epsilon\cdot\mathcal{O}((\frac{1}{|\mathcal{A}|})^2)$.

Proof: Consider a PreESS $\mathbf{X}_{\text{LB}}$ constructed by alternating two symbols, for example $0101\ldots01$. This PreESS has largest possible number of runs (n), and is composed of least symbol from the alphabet (2).

We describe a subset of arbitrary (ϵ,δ)-InDel patterns that result in a "large" $\mathbf{X}_{\text{LB}}$-post-edit set. In this subset of InDel patterns, we require that all the δn deletions precede all the ϵn insertions. Next, we require that the deletions, and then the insertions, occur in a "left-to-right manner" (so that a cursor, so to speak, first deletes all the locations to be deleted sequentially from left to right, and then starts from the beginning of the shortened sequence again to insert symbols in an analogous left-to-right manner). Further, the deletions may delete any δn non-pairwise-contiguous symbols (if a symbol is deleted, neither its two neighbor symbols will be deleted). Also each insertion may only insert symbols from $\{2,\ldots,|\mathcal{A}|-1\}$.

It can be verified that each 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. To do so, one first check for the "extra" symbols (those in the range $\{2,\ldots,|\mathcal{A}|-1\}$) to identify the insertion pattern uniquely. Then one takes out those "extra" symbols, aligns the remaining sequence to $\mathbf{X}_{\text{LB}}$ and checks for the "missing" symbols ($\{0,1\}$) to identify the deletion pattern uniquely (because no pairs of neighbor symbols got deleted). The overall InDel pattern is then the left-to-right composition of the deletion pattern and insertion pattern.

The number of such InDel patterns as described above is $\binom{n-\delta n}{\delta n}\binom{n-\delta n+\epsilon n}{\epsilon n}(|\mathcal{A}|-2)^{\epsilon n}$, hence is a lower bound on the number of PosESS $|\mathcal{Y}_{\epsilon,\delta}(\mathbf{X}_{\text{LB}})|$. The corresponding lower bound on the optimal achievable rate $R^*_{\epsilon,\delta}-\frac{1}{n}\log|\mathcal{Y}_{\epsilon,\delta}(\mathbf{X}_{\text{LB}})|$, is asymptotically $(1-\delta)H\left(\frac{\delta}{1-\delta}\right)+(1-\delta+\epsilon)H\left(\frac{\epsilon}{1-\delta+\epsilon}\right)+\epsilon\log\left(|\mathcal{A}|-2\right)$ by Stirling's approximation [24]. By expanding the binary entropy function and taking Taylor expansion,

$$(1-\delta)H\left(\frac{\delta}{1-\delta}\right)+(1-\delta+\epsilon)H\left(\frac{\epsilon}{1-\delta+\epsilon}\right)+\epsilon\log\left(|\mathcal{A}|-2\right) \tag{47}$$

$$=(1-\delta)\left(-\frac{\delta}{1-\delta}\log\frac{\delta}{1-\delta}-\frac{1-2\delta}{1-\delta}\log\frac{1-2\delta}{1-\delta}\right)+(1-\delta+\epsilon)\left(-\frac{\epsilon}{1-\delta+\epsilon}\log\frac{\epsilon}{1-\delta+\epsilon}-\frac{1-\delta}{1-\delta+\epsilon}\log\frac{1-\delta}{1-\delta+\epsilon}\right) \tag{48}$$

$$+\epsilon\log|\mathcal{A}|+\epsilon\log(1-\frac{2}{|\mathcal{A}|}) \tag{49}$$

$$=-\delta\log\frac{\delta}{1-\delta}-(1-2\delta)\log\frac{1-2\delta}{1-\delta}-\epsilon\log\frac{\epsilon}{1-\delta+\epsilon}-(1-\delta)\log\frac{1-\delta}{1-\delta+\epsilon}+\epsilon\log|\mathcal{A}|+\epsilon\log(1-\frac{2}{|\mathcal{A}|}) \tag{50}$$

$$=-\delta\log\delta-(1-2\delta)\log(1-2\delta)+(1-\delta)\log(1-\delta)-\epsilon\log\epsilon-(1-\delta)\log(1-\delta)+(1-\delta+\epsilon)\log(1-\delta+\epsilon) \tag{51}$$

$$+\epsilon\log|\mathcal{A}|+\epsilon\log(1-\frac{2}{|\mathcal{A}|}) \tag{52}$$

$$=H(\delta)+H(\epsilon)+\epsilon\log|\mathcal{A}|-(1-2\delta)\log(1-2\delta)+(1-\delta)\log(1-\delta)+(1-\epsilon)\log(1-\epsilon) \tag{53}$$

$$+(1-\delta+\epsilon)\log(1-\delta+\epsilon)+\epsilon\log(1-\frac{2}{|\mathcal{A}|}) \tag{54}$$

$$=H(\delta)+H(\epsilon)+\epsilon\log|\mathcal{A}|-(1-2\delta)(\log e)(-2\delta-\frac{(2\delta)^2}{2}-\mathcal{O}(\delta^3))+(1-\delta)(\log e)(-\delta-\frac{\delta^2}{2}-\mathcal{O}(\delta^3))+ \tag{55}$$

$$(1-\epsilon)(\log e)(-\epsilon-\frac{\epsilon^2}{2}-\mathcal{O}(\epsilon^3))+(1-\delta+\epsilon)(\log e)[-(\delta-\epsilon)-\frac{(\delta-\epsilon)^2}{2}-\mathcal{O}((\delta-\epsilon)^3)]+\epsilon(\log e)[-\frac{2}{|\mathcal{A}|}-(\frac{2}{|\mathcal{A}|})^2/2-\mathcal{O}((\frac{2}{|\mathcal{A}|})^3)] \tag{56}$$

$$=H(\delta)+H(\epsilon)+\epsilon\log|\mathcal{A}|+(\log e)(\epsilon^2-\delta^2-\epsilon\delta-\epsilon\frac{2}{|\mathcal{A}|})+\mathcal{O}(\max(\epsilon,\delta)^3)+\epsilon\cdot\mathcal{O}((\frac{2}{|\mathcal{A}|})^2) \tag{57}$$

$$\geq H(\delta)+H(\epsilon)+\epsilon\log|\mathcal{A}|-\frac{2}{|\mathcal{A}|}\epsilon-(2\log e)\max(\epsilon,\delta)^2+\mathcal{O}(\max(\epsilon,\delta)^3)+\epsilon\cdot\mathcal{O}((\frac{1}{|\mathcal{A}|})^2) \tag{58}$$

■

IV. Algorithm and Performance

We propose a unified coding scheme for both APES-AID and RPES-LtRRID processes. The coding scheme is a combination of dynamic programming (DP) 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 an expected description length that matches information-theoretic lower bounds up to lower order terms. Coding schemes achieving alphabet-size rates that match the lower bounds in Theorem 9 and Theorem 8 is an ongoing direnction.

A. Algorithm

For this section of a unified algorithm for both APES-AID and PRES-LtRRID processes, we unify the notation by notation without bars.

The encoder Φ_n takes in the following inputs: the PreESS $\mathbf{X}$ and the PosESS $\mathbf{Y}$, and outputs a transmission T as follows:
Step 1 DP-enc: The first subroutine of the encoder runs a *dynamic program* on the input $(\mathbf{X}, \mathbf{Y})$ to output an edit pattern $\tilde{\mathbf{E}}$ with $\tilde{\epsilon}n$ insertions and $\tilde{\delta}n$ deletions. This 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 typically run in time that is quadratic in n, the lengths of the strings being compared. We reference here Ukkonens work [25] since it gives an algorithm that is $\mathcal{O}(nk)$, where k refers to the *edit distance* – the minimum number of edits needed to process on $\mathbf{X}$ to get $\mathbf{Y}$, and is hence faster.
Step 2 Repre-enc: Represent the edit pattern $\tilde{\mathbf{E}}$ as a pair of sequences $(\tilde{O}^{n+\tilde{\epsilon}n}, \tilde{C}^{\tilde{\epsilon}n})$, where the edit operation pattern $\tilde{O}^{n+\tilde{\epsilon}n} \in \{\bar{\iota}, \bar{\Delta}, \bar{\eta}\}^{n+\tilde{\epsilon}n}$ specifies the edit operations of the output edit pattern by DP and the insertion content pattern $\tilde{C}^{\tilde{\epsilon}n} \in \mathcal{A}^{\tilde{\epsilon}n}$ specifies the content of insertions of the output edit pattern by DP.
Step 3 Entro-enc: The encoder uses Lempel-Ziv entropy code to compress $\tilde{O}^{n+\tilde{\epsilon}n}$ and $\tilde{C}^{\tilde{\epsilon}n}$.

The output of the encoder is a composition of the above three steps, $Enc(\mathbf{X}, \mathbf{Y}) = Entro(Repre(DP(\mathbf{X}, \mathbf{Y})))$.

The decoder decodes $\tilde{O}^{n+\tilde{\epsilon}n}$ and $\tilde{C}^{\tilde{\epsilon}n}$ by an entropy decoder corresponding to the entropy encoder in Step 3, and reconstructs $\mathbf{Y}$ from $(\mathbf{X}, \tilde{O}^{n+\tilde{\epsilon}n}, \tilde{C}^{\tilde{\epsilon}n})$.

B. Performance

It is well known in literature that *dynamic programming* finds the *edit distance* between two sequences – the minimal *total number* of edits (insertions, deletions and substitutions) needed to convert one sequence to the other. Whereas in our model with only insertions and deletions, it is straightforward to further deduce that the number of insertions and the number of deletions output by *DP* are both minimized, for the following reason. For all the edit patterns that converts $\mathbf{X}$ to $\mathbf{Y}$, the number of insertions (K_I) and the number of deletions (K_D) subject to the constraint $K_D - K_I = |\mathbf{X}| - |\mathbf{Y}|$, where the lengths of two source sequences $|\mathbf{X}|$ and $|\mathbf{Y}|$ are fixed given the two sequences. Hence, minimizing $K_D + K_I$ over all the edit patterns that converts $\mathbf{X}$ to $\mathbf{Y}$ minimizes both K_D and K_I. For the proof of Theorem 10 and 11, we only need a looser statement which is stated in the following Fact 3.

Fact 3. *The number of insertions (respectively the number of deletions) of the edit pattern output by dynamic programming $\tilde{\epsilon}n$ (respectively $\tilde{\delta}n$) is always no larger than the number of insertions of the actual edit pattern (respectively the number of deletions of the actual edit pattern). Hence, for the arbitrary (ϵ, δ)-Indel process,*

$$\tilde{\epsilon} \le \epsilon, \quad \tilde{\delta} \le \delta. \tag{59}$$

In the limit as the block length n goes to infinity, the compression rate of the above algorithm is $\lim_{n\to\infty} \frac{1}{n} H(\tilde{O}^{n+\tilde{\epsilon}n}, \tilde{C}^{\tilde{\epsilon}n})$. In the following we characterize upper bounds on the compression rate of the algorithm for both RPES-LtRRID process and APES-AID process.

1) Performance for RPES-LtRRID Process: In the RPES-LtRRID process, the number of deletions and insertions may exceed the expectation $\frac{\delta}{1-\epsilon}n$ and $\frac{\epsilon}{1-\epsilon}(n+1)$ respectively, in which case may lead to more bits transmitted. Moreover, the number of insertions can be unbounded. In Theorem 10 blow, we show that these events contribute a negligible amount to the achievable rate as the block length n tends to infinity, by using Chernoff bound to show that the probability the number of insertions/deletions is "much more" than expectation is exponentially small in block length n, while the amount contribute to the rate is polynomial in block length n.

Theorem 10. *The algorithm achieves a rate of at most $H(\delta) + H(\epsilon) + \epsilon \log|\mathcal{A}| + (\log|\mathcal{A}| + \log e - 2)\max(\epsilon, \delta)^{2-\tau} + \mathcal{O}(\max(\epsilon, \delta)^3)$ for any $tau > 0$ for the RPES-LtRRID process.*

Sketch proof: The number of deletions K_D is sum of n i.i.d. $\text{Bernoulli}(\frac{\delta}{1-\epsilon})$. Hence by Chernoff bound, $\Pr(K_D \ge (1+n^{-1/4})\frac{\delta}{1-\epsilon}n) \le e^{-\frac{\delta}{3(1-\epsilon)}\sqrt{n}}$. Similarly, the number of insertions K_I is the sum of $n+1$ i.i.d. $\text{Geo}_0(1-\epsilon)$. Hence by Chernoff bound, $\Pr(K_I \ge (1+n^{-1/4})\frac{\epsilon}{1-\epsilon}(n+1)) \le e^{-\frac{\epsilon}{3(1-\epsilon)}(\sqrt{n}+\frac{1}{\sqrt{n}})}$. Hence, with probability at least $1 - e^{-\frac{\delta}{3(1-\epsilon)}\sqrt{n}} - e^{-\frac{\epsilon}{3(1-\epsilon)}(\sqrt{n}+\frac{1}{\sqrt{n}})}$, by Fact 3, $\tilde{\delta} \le \frac{\delta}{1-\epsilon}(1+n^{-1/4})$ and $\tilde{\epsilon} \le \frac{\epsilon}{1-\epsilon}(1+n^{-1/4})(1+n^{-1})$. By Appendix C, the information rate contributes to $\lim_{n\to\infty}\frac{1}{n}H(\tilde{O}^{n+\tilde{\epsilon}n}, \tilde{C}^{\tilde{\epsilon}n})$ is at most $H(\frac{\delta}{1-\epsilon}) + H(\frac{\epsilon}{1-\epsilon}) + \frac{\epsilon}{1-\epsilon}\log\mathcal{A} + (\log e)(\frac{\epsilon}{1-\epsilon})^2 + \mathcal{O}\left((\frac{\epsilon}{1-\epsilon})^4\right) = H(\frac{\delta}{1-\epsilon}) + H(\frac{\epsilon}{1-\epsilon}) + \frac{\epsilon}{1-\epsilon}\log\mathcal{A} + (\log e)\epsilon^2 + \mathcal{O}(\epsilon^3)$.

With probability at most $e^{-\frac{\delta}{3(1-\epsilon)}\sqrt{n}}+e^{-\frac{\epsilon}{3(1-\epsilon)}(\sqrt{n}+\frac{1}{\sqrt{n}})}$, $K_D \in [(1+n^{-1/4})\frac{\delta}{1-\epsilon}n, n]$ and $K_I \in [(1+n^{-1/4})\frac{\epsilon}{1-\epsilon}(n+1)), n]$. The number of bits needed to specify the edit pattern is linear in n (bounded from the above by $2n + n\log\mathcal{A}$). However, the probability is exponentially small in n. Hence, as the block length n goes to infinity, the information contributed to $\lim_{n\to\infty}\frac{1}{n}H(\tilde{O}^{n+\tilde{\epsilon}n}, \tilde{C}^{\tilde{\epsilon}n})$ goes to zero.

The number of deletions K_D won't exceed n, whereas the number of insertions K_I can be unbounded. When K_I is larger than but still linear in n ($K_I = \Theta(n)$), the number of bits needed to specify the edit pattern is linear in n, whereas the probability of this event is exponentially small in n. Similarly, when $K_I = \Omega(n)$, the number of bits needed to specify the edit pattern is linear in K_I and the probability of is exponentially small in K_I. Hence, the amount of information rate contributes to $\lim_{n\to\infty}\frac{1}{n}H(\tilde{O}^{n+\tilde{\epsilon}n}, \tilde{C}^{\tilde{\epsilon}n})$ when the K_I exceeds n goes to zero as n goes to infinity.

From the above analysis, averaging over the randomness of the edit process, $\lim_{n\to\infty}\frac{1}{n}H(\tilde{O}^{n+K_I}, \tilde{C}^{K_I}) \leq H(\frac{\delta}{1-\epsilon}) + H(\frac{\epsilon}{1-\epsilon}) + \frac{\epsilon}{1-\epsilon}\log\mathcal{A} + (\log e)\epsilon^2 + \mathcal{O}(\epsilon^3)$. By Taylor expansion and the calculations below, the rate achieved by the algorithm is upper bounded by $H(\delta) + H(\epsilon) + \epsilon\log|\mathcal{A}| + (\log|\mathcal{A}| + \log e - 2)\max{(\epsilon,\delta)}^{2-\tau} + \mathcal{O}(\max{(\epsilon,\delta)}^3)$.

$$H(\frac{\delta}{1-\epsilon}) = -\frac{\delta}{1-\epsilon}\log\frac{\delta}{1-\epsilon} - \frac{1-\epsilon-\delta}{1-\epsilon}\log\frac{1-\epsilon-\delta}{1-\epsilon} \tag{60}$$
$$= -\frac{\delta}{1-\epsilon}\log\delta - \frac{1-\epsilon-\delta}{1-\epsilon}\log{(1-\epsilon-\delta)} + \log{(1-\epsilon)} \tag{61}$$
$$= -\delta(1+\epsilon+\mathcal{O}(\epsilon^2))\log\delta - (1-\epsilon-\delta)(1+\epsilon+\mathcal{O}(\epsilon^2))\log{(1-\epsilon-\delta)} + \log{(1-\epsilon)} \tag{62}$$
$$= [-\delta\log\delta - (1-\delta)\log{(1-\delta)}] - \delta(\epsilon+\mathcal{O}(\epsilon^2))\log\delta - (1-\delta+\mathcal{O}(\max(\epsilon,\delta)^2))\log{(1-\epsilon-\delta)} + \tag{63}$$
$$\log{(1-\epsilon)} + (1-\delta)\log{(1-\delta)} \tag{64}$$
$$= H(\delta) - \epsilon\delta\log\delta + (1-\delta+\mathcal{O}(\max(\epsilon,\delta)^2))(\log e)(\epsilon+\delta+(\epsilon+\delta)^2/2+\mathcal{O}((\epsilon+\delta)^3)) - \tag{65}$$
$$(\log e)(\epsilon+\epsilon^2/2+\mathcal{O}(\epsilon^3)) - (1-\delta)(\log e)(\delta+\delta^2/2+\mathcal{O}(\delta^3)) \tag{66}$$
$$= H(\delta) - \epsilon\delta\log\delta + (\log e)\cdot[\epsilon+\delta+\epsilon^2/2-\delta^2/2-\epsilon-\epsilon^2/2-\delta+\delta^2/2] + \mathcal{O}(\max(\epsilon,\delta)^3) \tag{67}$$
$$= H(\delta) - \epsilon\delta^{1-\tau} + \mathcal{O}(\max(\epsilon,\delta)^3) \tag{68}$$

$$H(\frac{\epsilon}{1-\epsilon}) = -\frac{\epsilon}{1-\epsilon}\log\frac{\epsilon}{1-\epsilon} - \frac{1-2\epsilon}{1-\epsilon}\log\frac{1-2\epsilon}{1-\epsilon} \tag{69}$$
$$= -\frac{\epsilon}{1-\epsilon}\log\epsilon - \frac{1-2\epsilon}{1-\epsilon}\log{(1-2\epsilon)} + \log{(1-\epsilon)} \tag{70}$$
$$= -\epsilon(1+\epsilon+\mathcal{O}(\epsilon^2))\log\epsilon - (1-2\epsilon)(1+\epsilon+\mathcal{O}(\epsilon^2))\log{(1-2\epsilon)} + \log{(1-\epsilon)} \tag{71}$$
$$= [-\epsilon\log\epsilon - (1-\epsilon)\log{(1-\epsilon)}] - \epsilon(\epsilon+\mathcal{O}(\epsilon^2))\log\epsilon - (1-\epsilon+\mathcal{O}(\epsilon^2))\log{(1-2\epsilon)} + (2-\epsilon)\log{(1-\epsilon)} \tag{72}$$
$$= H(\epsilon) - \epsilon^2\log\epsilon - (1-\epsilon+\mathcal{O}(\epsilon^2))(\log e)(-2\epsilon-(2\epsilon)^2/2+\mathcal{O}(\epsilon^3)) + (2-\epsilon)(\log e)(-\epsilon-\epsilon^2/2+\mathcal{O}(\epsilon^3)) \tag{73}$$
$$= H(\epsilon) - \epsilon^{2-\tau} + \mathcal{O}(\epsilon^3) \tag{74}$$

$$\frac{\epsilon}{1-\epsilon}\log\mathcal{A} = \epsilon(1+\epsilon+\mathcal{O}(\epsilon^2))\log\mathcal{A} \tag{75}$$
$$= \epsilon\log\mathcal{A} + (\log\mathcal{A})\epsilon^2 + \mathcal{O}(\epsilon^3) \tag{76}$$

■

2) Performance for APES-AID Process:

Theorem 11. *The algorithm achieves a rate of at most $H(\delta) + H(\epsilon) + \epsilon\log|\mathcal{A}| + (\log e)\epsilon^2 + \mathcal{O}(\epsilon^4)$ for the APES-AID process.*

Proof: The asymptotic compression rate of the algorithm in Section IV-A is $\lim_{n\to\infty}\frac{1}{n}H(\tilde{O}^{n+\tilde{\epsilon}n}, \tilde{C}^{\tilde{\epsilon}n}) = \lim_{n\to\infty}\frac{1}{n}H(\tilde{O}^{n+\tilde{\epsilon}n}) + \lim_{n\to\infty}\frac{1}{n}H(\tilde{C}^{\tilde{\epsilon}n})$ (the contents of insertions are independent with the positions of the edit operations). The empirical entropy of $\tilde{O}^{n+\tilde{\epsilon}n}$ can be calculated (in Appendix C), hence $\lim_{n\to\infty}\frac{1}{n}H(\tilde{O}^{n+\tilde{\epsilon}n}) = H(\tilde{\delta}) + H(\tilde{\epsilon}) + (\log e)\tilde{\epsilon}^2 + \mathcal{O}(\tilde{\epsilon}^4)$. The contents of insertions are uniformly drawn 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 the algorithm for the APES-AID process is at most $H(\tilde{\delta}) + H(\tilde{\epsilon}) + \tilde{\epsilon}\log|\mathcal{A}| + (\log e)\tilde{\epsilon}^2 + \mathcal{O}(\tilde{\epsilon}^4)$. By Fact 3, an upper bound of the compression rate is $H(\delta) + H(\epsilon) + \epsilon\log|\mathcal{A}| + (\log e)\epsilon^2 + \mathcal{O}(\epsilon^4)$. ■

APPENDIX A
DIFFERENT STOCHASTIC INDEL PROCESSES

There are potentially many ways to model a stochastic InDel process. In this paper, we study a left-to-right random InDel process modeled as a three-state Markov chain as shown in Fig. 1. It is a memoryless (i.i.d.) random InDel model. A more general left-to-right random InDel process with memory is shown in Fig. 2. More details are discussed in Section II-B1. The model was also studied in [8] as a channel with synchronization errors. The authors imposed a maximum insertion length, and the insertion/deletion probabilities to equal for the expected-length of the output sequence being the same as the input sequence. These two requirements are not needed in our paper. The authors in [8] 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 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. 14 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 the probability of the insertion-deletion interaction is of order $\mathcal{O}(\epsilon\delta)$, which to the lower order term. 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.

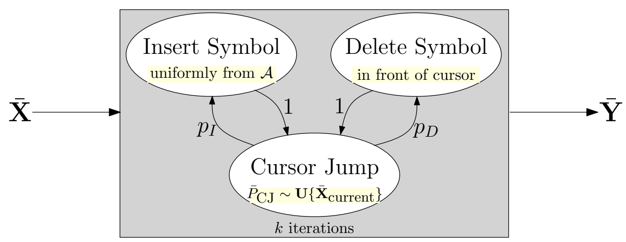

Fig. 14: other stochastic model 1

Since an insertion process can be regarded as the inverse of a deletion process, a random InDel process as in Fig. 15 was studied in [10]. The authors in [10] also considered the edit operation substitution. Here we hide the part corresponding to the substitution process to just represent the InDel process. In Fig. 15, 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 ($\mathrm{Binomial(n, 1-p_I)}$) sequence of i.i.d. symbols from $\mathcal{A}$. The authors in [10] 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 $\lim_{n\to\infty} \frac{1}{n} H(\bar{\mathbf{Y}}|\bar{\mathbf{X}})$.[10] 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[11]. Whereas one of our main effort was to characterize the explicit expression of the optimal rate.

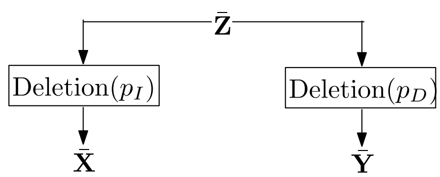

Fig. 15: other stochastic model 2

There are also many different stochastic insertion/deletion model in the line of works about 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 [13], [26]. In [26], capacity lower bounds for channels modeled as this InDel process are proposed. In [13], an algorithm for two-way file synchronization under non-binary non-uniform source alphabet was proposed. The Gallager model [27], also studied in [28], 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.

[10]Opposite from [10] in our paper we use $\bar{\mathbf{X}}$ for the side-information and $\bar{\mathbf{Y}}$ for the sequence to be synchronized.

[11]For the case with only deletions, the authors do have an information-theoretic lower bound in their earlier work [14]

APPENDIX B
PROOF OF FACT 1

We adopt the following notation in this proof:

1. Given a sequence, a newly inserted symbol is written with a superscript ι (α^ι).
2. Given a string, a deleted symbol is not actually deleted, but instead, is written with a subscript Δ (α_Δ).

Note that with this notation, the scenario of deleting an inserted symbol is represented as α^ι_Δ; the scenario of inserting a deleted symbol is represented as $\alpha_\Delta \alpha^\iota$.

Take PreESS $\mathbf{X}$ and perform the arbitrary (ϵ, δ)-InDel process , to obtain a string of length $m \leq n + \varepsilon n$ of which, at most δn symbols have Δ-subscript, and at most εnsymbols have ι-superscript.

We can discard symbols which have both Δ-subscript and ι-superscript (α^ι), and treat those as if they were never inserted in the first place. Since the symbols with only Δ-subscript are those found in the PreESS $\mathbf{X}$, it is obvious we can perform all the deletions first (an arbitrary δ-deletion process), and then all the insertion (an arbitrary $\frac{\epsilon}{1-\delta}$-insertion process because the ratio of number of insertions to the length of sequence after the deletions can be at most $\frac{\epsilon}{1-\delta}$) to obtain the exact same sequence.

APPENDIX C
ENTROPY ENCODING RATE OF $\tilde{O}^{n+\tilde{\epsilon}n}$

The entropy encoder Entro-enc encodes $\tilde{O}^{n+\tilde{\epsilon}n}$ at the empirical entropy. The empirical distribution of $\{\bar{\iota}, \bar{\Delta}, \bar{\eta}\}$ in $\tilde{O}^{n+\tilde{\epsilon}n}$ is

$$p_{\bar{\eta}} = \frac{1-\tilde{\delta}}{1+\tilde{\epsilon}}, p_{\bar{\iota}} = \frac{\tilde{\epsilon}}{1+\tilde{\epsilon}}, p_{\bar{\Delta}} = \frac{\tilde{\delta}}{1+\tilde{\epsilon}}. \tag{77}$$

The empirical entropy of the symbols $\{\bar{\iota}, \bar{\Delta}, \bar{\eta}\}$ in $\tilde{O}^{n+\tilde{\epsilon}n}$ is,

$$\lim_{n\to\infty} \frac{1}{(1+\tilde{\epsilon})n} H(\tilde{O}^{n+\tilde{\epsilon}n}) \tag{78}$$

$$= -\frac{1-\tilde{\delta}}{1+\tilde{\epsilon}} \log \frac{1-\tilde{\delta}}{1+\tilde{\epsilon}} - \frac{\tilde{\epsilon}}{1+\tilde{\epsilon}} \log \frac{\tilde{\epsilon}}{1+\tilde{\epsilon}} - \frac{\tilde{\delta}}{1+\tilde{\epsilon}} \log \frac{\tilde{\delta}}{1+\tilde{\epsilon}} \tag{79}$$

$$= \frac{1}{1+\tilde{\epsilon}} \cdot [H(\tilde{\delta}) + H(\tilde{\epsilon}) + (1-\tilde{\epsilon}) \log(1-\tilde{\epsilon}) + (1+\tilde{\epsilon}) \log(1+\tilde{\epsilon})] \tag{80}$$

$$\stackrel{(a)}{=} \frac{1}{1+\tilde{\epsilon}} \cdot [H(\tilde{\delta}) + H(\tilde{\epsilon}) + (1-\tilde{\epsilon})(\log e)(-\tilde{\epsilon} - \frac{\tilde{\epsilon}^2}{2} - \frac{\tilde{\epsilon}^3}{3} + \mathcal{O}(\tilde{\epsilon}^4)) + (1+\tilde{\epsilon})(\log e)(\tilde{\epsilon} - \frac{\tilde{\epsilon}^2}{2} + \frac{\tilde{\epsilon}^3}{3} + \mathcal{O}(\tilde{\epsilon}^4))] \tag{81}$$

$$= \frac{1}{1+\tilde{\epsilon}} \cdot [H(\tilde{\delta}) + H(\tilde{\epsilon}) + (\log e)\tilde{\epsilon}^2 + \mathcal{O}(\tilde{\epsilon}^4)], \tag{82}$$

where step (a) is by Taylor expansion.

Hence,

$$\lim_{n\to\infty} \frac{1}{n} H(\tilde{O}^{n+\tilde{\epsilon}n}) = H(\tilde{\delta}) + H(\tilde{\epsilon}) + (\log e)\tilde{\epsilon}^2 + \mathcal{O}(\tilde{\epsilon}^4). \tag{83}$$

REFERENCES

[1] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," *IDC iView: IDC Analyze the Future*, 2012.

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

[3] R. C. Burns and D. D. Long, "Efficient distributed backup with delta compression," in *Proc. Fifth Workshop on I/O in Parallel and Distributed Systems*, 1997, pp. 27–36.

[4] T. Suel and N. Memon, "Algorithms for delta compression and remote file synchronization," *Lossless Compression Handbook*, 2002.

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

[6] G. Cormode, M. Paterson, S. C. Sahinalp, and U. Vishkin, "Communication complexity of document exchange," in *Proc. of the ACM-SIAM Symp. on Discrete algorithms*, Jan. 2000.

[7] A. Orlitsky and K. Viswanathan, "Practical protocols for interactive communication," in *Proc. IEEE Int'l Symp. on Info. Theory*, 2001, p. 115.

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

[9] R. Venkataramanan, V. N. Swamy, and K. Ramchandran, "Efficient interactive algorithms for file synchronization under general edits," *arXiv preprint arXiv:1310.2026*, 2013.

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

[11] R. Venkataramanan, H. Zhang, and K. Ramchandran, "Interactive low-complexity codes for synchronization from deletions and insertions," in *Proc. 48th Allerton Conf. on Com., Control, and Comp.*, 2010.

[12] S. M. Yazdi and L. Dolecek, "Synchronization from deletions through interactive communication," in *Proc. IEEE Int. Symp. on Turbo Codes and Iterative Information Processing (ISTC)*, 2012, pp. 66–70.

[13] N. Bitouze and L. Dolecek, "Synchronization from insertions and deletions under a non-binary, non-uniform source," in *Proc. IEEE Int. Symp. on Information Theory Proceedings (ISIT)*, 2013, pp. 2930–2934.
[14] N. Ma, K. Ramchandran, and D. Tse, "Efficient file synchronization: A distributed source coding approach," in *Proc. IEEE Int. Symp. on Information Theory Proceedings (ISIT)*, 2011, pp. 583–587.
[15] S. E. Rouayheb, S. Goparaju, H. M. Kiah, and O. Milenkovic, "Synchronizing edits in distributed storage networks," *arXiv preprint arXiv:1409.1551*, 2014.
[16] Y. Kanoria and A. Montanari, "On the deletion channel with small deletion probability," in *Proc. IEEE Int. Symp. on Information Theory Proceedings (ISIT)*, 2010, pp. 1002–1006.
[17] A. Orlitsky, "Interactive communication of balanced distributions and of correlated files," *SIAM J. Discr. Math.*, vol. 6, no. 4, pp. 548–564, 1993.
[18] R. R. Varshamov and G. M. Tenenholtz, "A code for correcting a single asymmetric error," *Autom. Telemekh.*, vol. 26, pp. 288–292, 1965.
[19] V. I. Levenshtein, "Efficient reconstruction of sequences from their subsequences or supersequences," *Journal of Combinatorial Theory, Series A*, vol. 93, no. 2, pp. 310–332, 2001.
[20] V. Levenshtein, "Bounds for deletion/insertion correcting codes," in *Proc. IEEE Int. Symp. on Information Theory Proceedings (ISIT)*, 2002, p. 370.
[21] Y. Kanoria and A. Montanari, "Optimal coding for the binary deletion channel with small deletion probability," *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6192–6219, 2013.
[22] S. Ross, *A First Course in Probability 8th Edition*. Pearson, 2009.
[23] L. Ekroot and T. M. Cover, "The entropy of a randomly stopped sequence," *Information Theory, IEEE Transactions on*, vol. 37, no. 6, pp. 1641–1644, 1991.
[24] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.
[25] E. Ukkonen, "On approximate string matching," in *Foundations of Computation Theory*. Springer, 1983, pp. 487–495.
[26] E. Drinea and M. Mitzenmacher, "Improved lower bounds for the capacity of iid deletion and duplication channels," *Information Theory, IEEE Transactions on*, vol. 53, no. 8, pp. 2693–2714, 2007.
[27] R. G. Gallager, "Sequential decoding for binary channels with noise and synchronization errors," DTIC Document, Tech. Rep., 1961.
[28] M. Rahmati and T. Duman, "Bounds on the capacity of random insertion and deletion-additive noise channels," 2013.