Compression in a Distributed Setting*

Badih Ghazi[1], Elad Haramaty[2], Pritish Kamath[3], and Madhu Sudan[4]

1 **Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge MA 02139**
badih@mit.edu

2 **Harvard John A. Paulson School of Engineering and Applied Sciences**
seladh@gmail.com

3 **Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge MA 02139**
pritish@mit.edu

4 **Harvard John A. Paulson School of Engineering and Applied Sciences**
madhu@cs.harvard.edu

Abstract

Motivated by an attempt to understand the formation and development of (human) language, we introduce a "distributed compression" problem. In our problem a sequence of pairs of players from a set of K players are chosen and tasked to communicate messages drawn from an unknown distribution Q. Arguably languages are created and evolve to compress frequently occuring messages, and we focus on this aspect. The only knowledge that players have about the distribution Q is from previously drawn samples, but these samples differ from player to player. The only *common* knowledge between the players is restricted to a common prior distribution P and some constant number of bits of information (such as a learning algorithm). Letting T_ϵ denote the number of iterations it would take for a typical player to obtain an ϵ-approximation to Q in total variation distance, we ask whether T_ϵ iterations suffice to compress the messages down roughly to their entropy and give a partial positive answer.

We show that a natural uniform algorithm can compress the communication down to an average cost per message of $O(H(Q) + \log(D(P||Q))$ in $\tilde{O}(T_\epsilon)$ iterations while allowing for $O(\epsilon)$-error, where $D(\cdot||\cdot)$ denotes the KL-divergence between distributions. For large divergences this compares favorably with the static algorithm that ignores all samples and compresses down to $H(Q) + D(P||Q)$ bits, while not requiring $T_\epsilon \cdot K$ iterations that it would take players to develop optimal but separate compressions for each pair of players. Along the way we introduce a "data-structural" view of the task of communicating with a natural language and show that our natural algorithm can also be implemented by an efficient data structure, whose storage is comparable to the storage requirements of Q and whose query complexity is comparable to the lengths of the message to be compressed. Our results give a plausible mathematical analogy to the mechanisms by which human languages get created and evolve, and in particular highlights the possibility of coordination towards a joint task (agreeing on a language) while engaging in distributed learning.

1998 ACM Subject Classification E.4 Coding and Information Theory

Keywords and phrases Distributed Compression, Communication, Language Evolution, Isolating Hash Families

* The first author is supported in part by NSF STC Award CCF 0939370 and NSF Award CCF-1217423. The second author is supported in part by NSF Award CCF-1565641. The third author is supported in part by NSF CCF-1420956. The fourth author is supported in part by NSF Award CCF-1565641 and a Simons Investigator Award.

8th Innovations in Theoretical Computer Science (ITCS 2017).
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.

1 Introduction

Motivated by the goal of understanding human communication and in particular phenomena associated with the formation and development of language, we introduce a distributed compression problem and study it. We start with a description of the compression problem first, and then give our motivation.

1.1 Model

1.1.0.1 The Basic Model.

We consider a distributed setting where K players, with a complete network of point-to-point connections, are exchanging a sequence of messages drawn from an, apriori unknown, distribution Q. In our model the set of possible messages is a countable set, and we use $\mathbb{N}$, the set of natural numbers to denote this set without loss of generality. The communication proceeds in rounds: In round t, a message m is chosen from $\mathbb{N}$ according to Q independent of the past. Simultaneously an ordered pair of players $i, j \in [K] \overset{\text{def}}{=} \{1, \ldots, K\}$ with $i \neq j$ is chosen uniformly from all such pairs. The goal is for player i to encode the message m into a sequence of bits and send it to player j. Player j receives this sequence of bits and decodes it to a message $\hat{m}$. (Note that the encoding, and decoding, may depend on the history of interactions involving the sender, respectively receiver.) The round t is said to have an error if $m \neq \hat{m}$. The goal is to design encoding and decoding schemes that satisfy the condition that for every round t, the probability of error, over the history of random choices, is at most ε and the measure of performance is the expected length of communication averaged over the rounds up to t, studied as a function of t.

1.1.0.2 Efficiency Issues.

A second measure of performance of the encoding and decoding algorithms is their "computational efficiency". We define this notion using a "data-structural" perspective. Note that any encoder or decoder essentially needs to learn and store (approximations to) the distribution Q in order to perform moderately well. Thus, such an encoder or decoder needs to work with the amount of space that it might take to remember Q. At the same time, encoding or decoding a single message should not, and need not, take time linear in the storage. We thus measure the efficiency of the encoding and decoding algorithms in terms of its *space* requirement, and its *processing time* to compute the encoding of a message m including the time it takes to update its memory to incorporate this new message in its history.

1.1.0.3 Setup Assumptions.

Finally, we parameterize one commonality in the initialization of the different players. Note that to initialize any communication the players must have some way of exchanging messages. One may consider the natural binary description of messages as one such possibility. Other possibilities may go via Kolmogorov complexity, i.e., by letting the players share a common universal machine and then representing a message via the encoding of machine that outputs the binary representation of the message and halts. Rather than choosing any one of these representations, we parametrize the setup by the exact initial representation. More precisely, we consider a distribution P on $\mathbb{N}$ for which a given initial representation is optimal and assume that all players share P at the outset[1]. Thus the encoding and decoding algorithms may depend on this prior distribution, but otherwise the algorithms must be completely uniform and may not rely on any other shared information. Note that Q is chosen adversarially and has no relationship to P, however we will allow our performance, i.e., the expected length of the compression to depend on the gap (distance or divergence) between P and Q.

1.2 Motivation: Language Formation and Development

Our motivation to study this distributed compression problem is to give a fresh perspective on phenomena associated with the formation and evolution of human languages. We note that the study of languages is a central quest in linguistics, cognitive science and philosophy and much is known about it based on empirical studies. Our hope is to add some mathematical flavors to this.

For our purpose, we may view language as providing a map that describes how to convert a message in an individual's brain into a sequence of utterances. Yet no language has a short description of this map. Part of the challenge seems to be that language is constantly evolving and if one were to fix any bound, language seems to evolve to a point where the description length exceeds this bound. The reason for this evolution may be viewed as some form of compression. While the ultimate goal may not be the time it takes to convey a message, language certainly evolves by creating shortcuts for currently frequent messages[2]. This motivates our use of the compression capability of the message-to-utterance map as a crude measure of performance. It is not the unique goal, but it is well-aligned with the goals of language.

A second feature about languages is that no two individuals probably have identical descriptions of the map. Attempts to give a unified description of the language (say, as in a dictionary) end up with many different dictionaries and each one capturing some segment of the population. Yet, language is robust to this variation and for the most part, communication manages to work despite the lack of agreement on the dictionary. Our view of this diversity is to consider the process of language acquisition. Individuals (children) learn from examples and indeed there is major diversity in the set of examples one encounters depending on one's own circumstances, but even if one were to factor out this diversity (e.g., by considering identical twins), their experiences are still different. This inspires our setting: individuals are all born identical and get samples from the same distribution. (Furthermore there are no network effects - the underlying graph is a complete graph and the message distribution is independent of the edge distribution. We will discuss this shortly.) Yet their samples are not identical and even this minor discrepancy seems to foil simple algorithms to coordinate on a compression map and introduces either diversity in the map, or complexity in the coordination process. Thus, the distributed compression problem already gives a potential reason for the diversity in language.

We emphasize that our choice of a simple graph (the complete graph) and the independence between the messages and the graph are not restrictions of the "model". It is quite easy to extend our model to the setting where the graphs are complex, the distributions on edges are weighted and to allow the distribution of the messages to depend on the edge. While

[1] Intuitively, we can think of P as being a primitive "gesturing language" that is understandable to all people.

[2] For instance, a language could evolve to use the word "Ix" to denote "a boy who is not able to satisfactorily explain what a Hrung is, nor why it should choose to collapse on Betelgeuse Seven" [1].

such richness is permitted by the model, we restrict to the simple setting to allow simpler contrasts between basic options (and our more sophisticated one).

Finally, one intriguing aspect of language is the amount of influence that different players have on its development. For the most part, language evolution seems to be a decentralized process, but this does not imply equal influence for all players. The role of books, especially those on grammar or dictionaries, of the media, and popular figures definitely assigns disproportionate influence to different players. A question that might be asked is whether language could manage to gain coherence across the population in the absence of such highly influential figures. Our model offers a way to study such questions (in our simplified setting of compression).

1.3 Context and Main Benchmarks

Our main result is a distributed compression algorithm with "decent" performance. To set the stage for this algorithm, we first describe some basic benchmarks and then some basic compression schemes.

In what follows, we use $Q(m)$ to denote the probability of a message m being drawn according to distribution Q. We let $H(Q) = \sum_{m\in\mathbb{N}} Q(m)\log_2(1/Q(m))$ denote the binary entropy of Q and we let $D(Q||P) = \sum_{m\in\mathbb{N}} Q(m)\log_2(Q(m)/P(m))$ denote the KL-divergence between Q and P. The best possible compression scheme would need at least $H(Q)$ bits per message in expectation — this is true in the 2-person case and we will discuss below whether this is achievable in the distributed setting.

We refer to $\tau = 2t/K$ as the *local time*, which roughly measures the number of messages any one player has seen (either as sender or receiver) at time t. We use T_ε to denote the local time by when a fixed player can obtain a ε-close approximation to Q, with probability at least $1-\varepsilon$. Note that T_ε can be upper bounded by $O(2^{H(Q)/\varepsilon})$ (and so in particular T_ε is finite for distributions with finite entropy). Intuitively, T_ε is a reasonable measure of local time by which one may expect to be able to compress well according to Q (even in the simple 2-player setting) and this will be a benchmark time for our compression algorithms also.

Finally, a natural upper bound on the space complexity of storing (an ε-approximation to) Q is again $2^{H(Q)/\varepsilon}$. We will compare the storage needs of the various solutions below to this benchmark. Natural measures of update times would be polylogarithmic in space and we will ask for that. (In what follows, we assume messages are given as black boxes that can be stored in unit time and space and that basic operations such as comparison of messages (is $m_1 \leq m_2$?) take unit time.)

We now turn to some basic schemes for compression.

Near Ideal Compression We first point out the (obvious?) flaw with the most natural hope one may have: Players could try to learn Q and get ε-close to the right distribution moderately fast (in local time T_ε) and then use the optimal (Huffman) coding applied to such a distribution. Unfortunately, they can not agree on this naive distribution and so no naive variation of the 2-player compression mechanism seems to be implementable.

Static Compression: Players simply encode and decode according to the Huffman code for distribution P. The error probability is zero and the expected length of the compression will be at most $H(Q) + D(Q||P) + O(1)$. The good news with this scheme is that the performance does not depend on K, but the bad news is that players do not learn to speak more effectively from examples. This is captured by the fact that the gap from optimal compression is $D(Q||P)$ and we think of this as a large quantity.

Point-to-Point Compression: For every ordered pair (i, j), player i uses the Lempel-Ziv (or any universal) compression algorithm restricted to the sequence of messages that were directed from i to j and player j decodes according to the same history. This scheme converges to a compression length of $H(Q)$ but it takes a relatively long time - a local time of $K \times T_\varepsilon$. We view dependence on K in the local time as too high. This scheme also involves memory requirement which is K times larger than the space needed for a 2-player solution.

Dictatorial Compression: Here one player (the dictator) is singled out and tasked with the compression task. He learns a distribution close to Q and then communicates the resulting encoding/decoding scheme to all other players. The compression achieved by this scheme is near-optimal (converges to $H(Q)$); and the space requirement is also near-optimal. The main quantitative weakness we see is a mild dependence on K in the time it takes for this scheme to converge: Specifically it takes about T_ε local time for the dictator to learn the distribution (which is perfectly fine), but then it needs to spread the information out to all K players and this takes $T_\varepsilon + \Theta(\log K)$ additional local time (using any reasonable gossip algorithm with proper pipelining of messages). The main "criticism" of the scheme may be that it is centralized. While centralized mechanisms do plausibly play a role in the development of languages, they do not seem to be the only mechanism, and so we seek a truly distributed solution below.

1.4 Results

We now state our main theorem.

▶ **Theorem 1** (Main Theorem). *Let $\varepsilon > 0$ be a sufficiently small positive absolute constant. For all K and P, there exists a deterministic distributed compression protocol Π, such that for any distribution Q over $\mathbb{N}$ when run for T iterations,*

- *the amortized communication cost of Π over T iterations approaches $O(H(Q)+\log D(Q||P)+\log(1/\epsilon))$ as T gets large. More formally, the amortized communication cost is*

$$O\left(H(Q) + \log D(Q||P) + \log(1/\epsilon) + \frac{2^{\Theta(H(Q)+D(Q||P))/\varepsilon} \cdot K}{T} \cdot D(Q||P) + 1\right)$$

- *in each round, the transmitter and receiver run in time linear in their input and output sizes.*
- *the space usage is exponential in $(H(Q) + D(Q||P))/\epsilon$.*

Our scheme is obtained with each player mixing the static scheme (used initially) with a switch to a more complex scheme once a sufficiently good approximation to Q has been learned (by the player). A central ingredient in our scheme is a solution to the "Uncertain Compression" problem studied by Juba et al. [6] and Haramaty and Sudan [4]. In the uncertain compression problem, two players attempt to compress a single message drawn from a distribution Q, but only the sender knows P and the receiver only knows some distribution Q' which is close to Q. The uncertain compression problem seems to arise naturally in our setting (neither the sender nor the receiver know Q in our case, but both are close and this mild difference can simply be ignored). [6] give a "randomized" solution to this problem which compresses messages roughly down to $H(Q) + O(1)$ bits. Adapting this solution to our setting, essentially as a black box, achieves similar effects in our setting (compression down to $H(Q) + \delta \cdot D(Q||P)$ bits in time $\delta^{-1} \cdot T_\varepsilon$ local time), but a flaw with this scheme is that it requires the players to share a large random string in the setup phase. Instead we turn to the solution of [4] which does not need any randomness, but their solution assumes that Q is supported on a finite set (of size N) and their compression length is $O(H(Q) + \log\log N)$. Since our distributions are not supported on finite sets, we need to modify their scheme and a careful modification followed by a relatively straightforward analysis leads to our eventual scheme and analysis. In the process, we are also able to build a small data structure implementing the encoding and decoding with efficient processing time. We point out that if one does not care about computational efficiency, then we can remove the additive $\log(1/\epsilon)$ term from the communication cost in Theorem 1 above while also replacing the multiplicative $2^{\Theta(H(Q)+D(Q||P))/\varepsilon}$ factor by $2^{\Theta(H(Q))/\varepsilon}$ (for more details, see Section 3.6).

1.5 Previous Work on Language Evolution

There have been many works on language evolution (to the best of our knowledge all from outside the theoretical computer science community). Without trying to be exhaustive, we briefly mention some of them. In the lingustics field, significant work has been done in the last decades on trying to understand language evolution, including [2, 3]. Several papers also study language from the landscape of evolutionary game theory and evolutionary biology, e.g., [14, 12, 13, 9, 10, 11, 5, 7, 8], and neuroscience, e.g., [16]. There has also been some previous attempts to connect language evolution to the framework of information theory (e.g., [15]), but their focus is on word formation in the "two-player" case, unlike our setup where we consider language as the outcome of the interaction between several players. To the best of our knowledge, the distributed compression perspective developed in this paper has not been considered before.

1.6 Conclusions

We believe that the model raised is an extremely interesting one and is quite pertinent to the analyses of collective distributed phenomena where distributed entities are trying to come together to form joint actions. We believe the process and notation permit a much richer study, especially when one starts to allow correlations between the messages generated and the sender-receiver pairs. The ability to study the encoding and decoding functions — are they really functions, are they inverses of each other, how do they evolve? — are all intriguing questions that can now be subject to analyses. While our results do not address all these aspects, we do hope it will be the subject of future work.

In terms of the constructions and results, one interesting aspect of the compression protocol we use is that it mimics some of the curious features shown in human language. For every message m, player i and round t, the encoding function describes a specific word which is player i's encoding of m, i.e., it gives a (encoding) dictionary. The same player also possesses at the same round a decoding dictionary which we may view as saying, for every message m, which words this player would decode to m. Unlike in the basic schemes described, in our scheme the encoding dictionary is not identical to the decoding dictionary. While the encoding dictionary is a function mapping messages to words, the decoding dictionary is not: It is more conservative and lists many words for any given message. This phenomenon is definitely visible in human languages and our work suggests a plausible reason for the occurence of this phenomenon.

We now mention some important questions that arise from this work. On the conceptual side, it would be very interesting to further use the formalism and ideas developed in theoretical computer science over the last decades in order to capture the phenomena exhibited by human languages. In particular, it would be interesting to extend our model to take into account other objectives along with compression. It would also be very interesting to consider the case where Q and the set of interacting players vary (slightly) with time, in the hope of modelling cultural changes that take place from one generation to another.

On the more technical side, we sticked in this work to the complete graph representing the interactions between various players. It would be worthwile to investigate other graph structures that favor the creation of communities, and study the properties of the language(s) that evolve in this case. Moreover, while we have considered in this work generic distributions Q, it would be nice to explore the data-structural aspects in the case where Q comes from a well-structured family of distributions (e.g, a Markov Chain). Finally, a concrete question is to determine whether the $O(\log(D(P||Q)))$ additive term in Theorem 3 is actually needed, which seems to be related to some intriguing questions about the chromatic number of certain families of graphs (see [4]).

1.6.0.1 Outline of the Rest of the Paper

In Section 2, we formally define our distributed compression model. In Section 3, we describe our main protocol along with its computationally efficient implementation (Section 3.1, Section 3.2, Section 3.3, Section 3.4 and Section 3.5). In Section 3.6, we describe a computationally inefficient variant of our protocol that requires smaller communication.

2 Formal Definitions

Throughout this paper, we denote by $H(Q) \triangleq \sum_x Q(x) \log(1/Q(x))$ the Shannon entropy of a probability distribution Q, and by $D(Q||P) \triangleq \sum_x Q(x) \log(Q(x)/P(x))$ the KL divergence between probability distributions Q and P. For any set S of elements, we write $u \in_R S$ to mean that u is sampled uniformly at random from the set S. We also denote by $\mathbb{N}$ the set of all natural numbers.

We now formally define our setup.

▶ **Definition 2** (Distributed Compression). A *distributed compression protocol* Π is parametrized by a tuple (K, P, ε) where

- K is the number of players.
- P is a prior distribution over $\mathbb{N}$, which the players all agree on.
- ε is an error parameter.

The protocol is run on an instance parametrized by a pair (Q, T) where Q is the “true” distribution over $\mathbb{N}$, and T is the total number of iterations for which the protocol is run. Both Q and T are unknown to the players. In any iteration $t \in [T]$,

- Two distinct players i and j are chosen uniformly at random from $[K]$.
- A message m is sampled from distribution Q, and is given to player i.
- Player i attempts to communicate m to player j by sending a single message comprising of C_t bits.
- Player j outputs a message $\widehat{m}$.

The protocol is required to be such that, for any Q, and in any iteration t, it holds that $\Pr[\widehat{m} \neq m] \leq \varepsilon$, where the the probability is over the randomness of the messages and players chosen in the history of the protocol. The amortized communication cost of Π is defined to be $\sum_{t\in[T]} C_t/T$.

During the description of the protocol and the analysis, we will use t to denote the current iteration. Also, we will use t_i to denote the 'local time' of player i. That is, t_i is the number of times player i was picked as the sender. Note that $t = \sum_{i \in [K]} t_i$.

3 A Distributed Compression Protocol

In this section, we prove the following theorem, which is the same as Theorem 1 but "without the computational efficiency" part. The proof of the computational efficiency part of Theorem 1 appears in Section 3.5.

▶ **Theorem 3.** *Let $\varepsilon > 0$ be a sufficiently small positive absolute constant. For all K and P, there exists a deterministic distributed compression protocol Π, such that for any distribution Q over $\mathbb{N}$ when run for T iterations, the amortized communication cost of Π over T iterations approaches $O(H(Q) + \log D(Q||P) + \log(1/\epsilon))$ as T gets large.*

More formally, for $T \geq 8 \cdot 2^{2H(Q)/\varepsilon} \cdot \log(8/\varepsilon) \cdot K$, the amortized communication cost is

$$O\left(H(Q) + \log D(Q||P) + \log(1/\epsilon) + \frac{2^{2H(Q)/\varepsilon} \cdot \log(8/\varepsilon) \cdot K}{T} \cdot D(Q||P) + 1\right).$$

In the rest of this section, we describe the protocol behind the proof of Theorem 3.

3.1 Overview of the Protocol

We begin by giving a brief overview of the protocol. In any iteration, the chosen players will use one of two protocols that we call Static protocol (Section 3.2) and Uncertain protocol (Section 3.3).

On a high level, the Static protocol communicates messages with zero error, but it uses $H(Q) + D(Q||P) + O(1)$ bits of communication in expectation. On the other hand, the Uncertain protocol communicates $O(H(Q) + \log(D(Q||P)))$ bits in expectation, but it makes errors with some probability.

Suppose during iteration t, a message m is chosen to be sent by player i to player j, where m is sampled from the unknown distribution Q. In this case, player i will decide to communicate using either the Static protocol or the Uncertain protocol. Intuitively, in the initial few rounds in which player i is the sender, she will use the Static protocol as she does not want to risk incurring large error by using the Uncertain protocol. But, once player i has seen enough messages, she will switch to using the Uncertain protocol. The final bound on the amortized communication cost comes about by showing that the protocol ends up using the Uncertain protocol much more often than the Static protocol.

In Section 3.4, we describe exactly how the players switch between the two protocols and prove Theorem 3.

3.2 The Static Protocol

In the Static protocol, player i uses the Huffman codebook for distribution P in order to communicate the message m. The expected communication cost of doing so is $H(Q) + D(Q||P) + O(1)$. The good aspects of this protocol are that the error probability is zero, and the players do not require any knowledge about the unknown distribution Q. However, the downside is that the communication cost is quite high in terms of the dependence on $D(Q||P)$.

We summarize the Static protocol in the following straightforward lemma.

▶ **Lemma 4** (Static Protocol). *Suppose that during iteration t, a message m is chosen to be sent by player i to player j, where m is sampled according to the unknown distribution Q. Then, player i can communicate m to player j with zero error, such that the expected communication length is upper bounded by*

$$H(Q) + D(Q||P) + 1.$$

3.3 The Uncertain Protocol

The Uncertain protocol is suitable when the players have individually learnt good estimates of the distribution Q. However, since the players do not exactly agree on their learned estimates, we need an approach for the players to communicate when their estimates of Q are *close* but may not be exactly identical. Our approach is inspired from [4], and we obtain a protocol that in expectation communicates roughly $O(H(Q) + \log D(Q||P)) + O(1)$ bits. We summarize the Uncertain protocol in the following lemma,

▶ **Lemma 5** (Uncertain Protocol). *Suppose that during iteration t, a message m is chosen to be sent by player i to player j, where m is sampled according to the unknown distribution Q. Then, player i can communicate m to player j, such that the expected communication length is upper bounded by*

$$O\left(H(Q) + \log D(Q||P) + \log(1/\epsilon) + 1\right).$$

Moreover, the error probability is at most

$$2 \cdot e^{-\frac{1}{8}\frac{Q(m)}{K}t} + \frac{\epsilon}{4},$$

where the randomness is over all past messages and players chosen in the previous iterations.

3.3.0.1 Isolating Hash Families

In order to describe the Uncertain protocol achieving Lemma 5, we will need the following notion of an *isolating hash family* which generalizes that of [4].

▶ **Definition 6** (Isolating Hash Families). Let N, R and ℓ be positive integers and $\varepsilon \in (0,1]$. Then, a collection $\mathcal{H} = \{h_1, h_2, \dots, h_M : [N] \to [R]\}$ is said to be (N, ℓ, ϵ)-isolating if for every subset $S \subseteq [N]$ with $|S| \le 2^{\ell-1}$ and every $m \in [N] \backslash S$, we have that $\Pr_{h\in\mathcal{H}}[h(m) \in h(S)] < \varepsilon$. We call M the *size* and R the *range-size* of the isolating hash family $\mathcal{H}$. The family $\mathcal{H}$ is said to be *efficiently computable* if there is an algorithm that takes as input $i \in [M]$ and $j \in [N]$ and computes $h_i(j)$ in time polynomial in $\log M$, $\log N$ and $\log R$.

We note that the family used in [4] corresponds to setting $\epsilon = 1$ in Definition 6. The next lemma shows the existence of an *explicit* and *efficiently computable* (N, ℓ, ϵ)-isolating hash family of relatively small size and small range-size.

▶ **Lemma 7.** *For every positive integers N and ℓ and every $\varepsilon \in (0,1]$, there exists an* explicit *and* efficiently computable *(N, ℓ, ε)-isolating hash family $\mathcal{H}_{(N,\ell,\varepsilon)}$ of size and range-size at most $2^{\ell} \cdot \frac{\log N}{\varepsilon}$.*

Proof. Let $q = 2^{\ell + \lceil \log n + \log \frac{1}{\varepsilon} \rceil}$. For each $x \in \mathbb{F}_q$, define the function h_x to be the evaluation of the polynomial defined by m on x, i.e.,

$$h_x(m_0, ..., m_{n-1}) \triangleq \sum_{i=0}^{n-1} m_i x^i.$$

By the fundamental theorem of algebra, for every $m' \neq m$, we have that $\Pr_x[h_x(m) = h_x(m')] \leq \frac{n}{q} \leq 2^{-\ell - \log \frac{1}{\varepsilon}}$. Thus, by the union bound, for every set S of size at most $2^{\ell-1}$, we have $\Pr[f(m) \in f(S)] \leq \varepsilon$, as required. ◀

3.3.0.2 Pre-Processing Step

As stated earlier, all the players come in with a prior distribution P. In addition, as part of the pre-processing, they compute and store the following:

- Divide the input space $\mathbb{N}$ into a countable number of buckets indexed by $r \in \mathbb{N}_{>0}$, given by $A_r = \left\{m : 2^{-r} < P(m) \leq 2^{-r+1}\right\}$. Clearly, for any r, it holds that $|A_r| \leq 2^r$. In addition, define the function $r(m) := \lceil \log(1/P(m)) \rceil$ for every $m \in \mathbb{N}$, that is, $r(m)$ is the index of the bucket to which m belongs.
- For every r, fix an (arbitrary) choice of isolating hash families $\mathcal{H}_{(N,\ell,\epsilon/4)}$, for $N = |A_r|$ and every choice of $\ell \in \{1, 2, \cdots, \lceil \log N \rceil\}$.

Suppose during iteration t, a message m is chosen to be sent by player i to player j, where m is sampled according to the unknown distribution Q. Define Q_t^i to be the empirical distribution of the samples seen by player i up to iteration t (which includes the iteration t, where the message seen is m). Similarly, define Q_t^j to be the empirical distribution of the samples seen by player j up to iteration t (this includes iteration t, but by definition player j does not see any message in this iteration). The players use the encoding and decoding strategies described next.

3.3.0.3 Encoding

Upon receiving message m, player i does the following,

(i) let $A \overset{\text{def}}{=} A_{r(m)}$ and $N \overset{\text{def}}{=} |A|$.

(ii) let $\ell = \lceil \log(4/Q_t^i(m)) \rceil$.

(iii) let $u \in_R [\mathcal{H}_{(N,\ell,\epsilon/4)}]$.

(iv) Send the tuple $(r, \ell, u, h_u(m))$ to player j.

The intuition for this encoding is as follows: upon receiving r, player j understands that $m \in A_r$, upon receiving ℓ, she understands which hash family to use, upon receiving u, she knows which hash function to use, and hopefully with $h_u(m)$, she will be able to recover m correctly.

3.3.0.4 Decoding

Upon receiving the tuple (r, ℓ, u, h_*), player j does the following:

(i) Set $A = A_r$ and $N \overset{\text{def}}{=} |A|$.

(ii) Identify $h_u \in \mathcal{H}_{(N,\ell,\epsilon/4)}$.

(iii) Output $\arg\max_{m' \in A : h_u(m') = h_*} Q_t^j(m')$.

3.3.1 Analysis

We now analyze the operation of the above protocol.

3.3.1.1 Communication Cost

Suppose the message m is chosen to be sent by player i to player j. The communication cost of sending the tuple $(r, \ell, u, h_u(m))$ is as follows:

(i) $\log \lceil \log(1/P(m)) \rceil$ bits to send r.

(ii) $\log(\log(1/Q_t^i(m)) + 3)$ bits to send ℓ, since $\ell \le \log|S| + 1 \le \log(4/Q_t^i(m)) + 1$.

(iii) $\log(1/Q_t^i(m)) + \log \lceil \log(1/P(m)) \rceil + \log(1/\epsilon) + 5$ bits to send u (it takes $\ell + \log\log N + \log(4/\epsilon)$ bits).

(iv) $\log(1/Q_t^i(m)) + \log \lceil \log(1/P(m)) \rceil + \log(1/\epsilon) + 5$ bits to send $h_u(m)$.

Thus, the total communication is given by,

$$2\underbrace{\log(1/Q_t^i(m))}_{(I)} + 3\underbrace{\log \lceil \log(1/P(m)) \rceil}_{(II)} + \underbrace{\log(\log(1/Q_t^i(m)) + 3) + 10}_{(III)} + 2\log(1/\epsilon)$$

We wish to prove guarantees on the expected communication cost, when m is drawn from Q. The terms in (III) are lesser order terms, which are smaller than (I), thus we can ignore them. Term (II) in expectation is,

$$\mathop{\mathbb{E}}_{m\sim Q}\left[\log\left(\left\lceil \log\frac{1}{P(m)}\right\rceil\right)\right] \;\le\; \log\left(\mathop{\mathbb{E}}_{m\sim Q}\left\lceil \log\frac{1}{P(m)}\right\rceil\right) \;\le\; \log(H(Q) + D(Q||P) + 1)$$

Term (I) is slightly more tricky to bound in expectation. Note that the empirical distribution changes on receiving message m (this turns out to be critical in bounding the communication!). That is, $Q_t^i(m) = \frac{1+(t-1)Q_{(t-1)}^i(m)}{t}$. Also let $\mathcal{M}_t^i$ be the multi-set of all messages that player i has seen up to time t. Thus, Term (I) in expectation is as follows,

$$\mathop{\mathbb{E}}_{\mathcal{M}_{(t-1)}^i}\ \mathop{\mathbb{E}}_{m\sim Q}\left[\log\frac{1}{Q_t^i(m)}\right] \;=\; H(Q) + \mathop{\mathbb{E}}_{\mathcal{M}_{(t-1)}^i}\ \mathop{\mathbb{E}}_{m\sim Q}\left[\log\frac{Q(m)}{\frac{1}{t_i} + \frac{(t_i-1)Q_{(t-1)}^i(m)}{t_i}}\right]$$

In order to bound the second term above, we consider two cases, (i) $Q_{(t-1)}^i(m) \ge Q(m)/2$ or (ii) $Q_{(t-1)}^i(m) < Q(m)/2$. After fixing t_i and m, by Chernoff bound over the randomness of $\mathcal{M}_{(t-1)}^i$ we have that case (i) happens with probability at least $1 - \exp(-t \cdot Q(m)/8)$.

$$\text{Case (i) } Q_{(t-1)}^i(m) \ge Q(m)/2 \quad\Longrightarrow\quad \log\left(\frac{Q(m)}{\frac{1}{t_i} + \frac{(t-1)Q_{(t)}^i(m)}{t_i}}\right) \le 1$$

$$\text{Case (ii) } Q_{(t-1)}^i(m) < Q(m)/2 \quad\Longrightarrow\quad \log\left(\frac{Q(m)}{\frac{1}{t_i} + \frac{(t_i-1)Q_{(t-1)}^i(m)}{t_i}}\right) \le \log(t_i \cdot Q(m))$$

Using these upper bounds we get that,

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathcal{M}_{(t-1)}^i}\ \mathop{\mathbb{E}}_{m\sim Q}\left[\log\frac{Q(m)}{\frac{1}{t} + \frac{(t_i-1)Q_{(t-1)}^i(m)}{t_i}}\right] &\le \mathop{\mathbb{E}}_{m\sim Q}\left[1\cdot\left(1 - e^{-t_i\cdot Q(m)/8}\right) + \log(t_i\cdot Q(m))\cdot e^{-t_i\cdot Q(m)/8}\right] \\
&\le 1 + \mathop{\mathbb{E}}_{m\sim Q}\left[\log(t_i\cdot Q(m))\cdot e^{-t_i\cdot Q(m)/8}\right] \\
&\le 2\,,
\end{aligned}$$

where the last inequality just follows from the fact that $\log(x) \cdot e^{-x/8} \le 1$ for all x.
Thus the overall communication is bounded by

$$(2 + o(1))\, H(Q) + 3\log D(Q||P) + 2\log(1/\epsilon) + O(1)\,.$$

3.3.1.2 Error Guarantee

We now show that the error probability in iteration t, denoted by p_t^{err} of the protocol is upper bounded by $2 \cdot e^{-\frac{1}{8}\frac{Q(m)}{K}t} + \epsilon/4$, where m is fixed to be the message sent in round t.

Since player i has communicated (r, ℓ, u), player j knows the correct bucket of messages A_r to which m belongs. Knowing ℓ and u, player j also knows which hash function is being used, which is chosen to ensure that for every set S of size $\leq 2^{\ell}$, with probability $1 - \varepsilon/4$, for all $m' \in S \setminus \{m\}$, $h_u(m) \neq h_u(m')$.

Thus, if $\ell \leq \log(1/Q_t^j(m))$ then the j-th player will distinguish m from the set $S = \{m' \in A \mid Q_t^j(m') \geq Q_t^j(m)\}$ with probability $1 - \varepsilon/4$. We will bound the probability that this does not happen.

$$\begin{aligned}
p_t^{\text{err}} &\leq \Pr\left[\ell > \log(1/Q_t^j(m))\right] + \frac{\epsilon}{4} \\
&\leq \Pr\left[4 \cdot Q_t^i(m) \geq Q_t^j(m)\right] + \frac{\epsilon}{4} \\
&\leq \Pr\left[Q_t^i(m) \geq 2 \cdot Q(m)\right] + \Pr\left[Q_t^j(m) \leq \frac{1}{2}Q(m)\right] + \frac{\epsilon}{4} \\
&\leq e^{-\frac{1}{3}\frac{Q(m)}{K}t} + e^{-\frac{1}{8}\frac{Q(m)}{K}t} + \frac{\epsilon}{4} \, ,
\end{aligned}$$

where the last equality follows by Chernoff bound and the fact that Q_t^i, Q_t^j are binomial distributions with parameters t and $\frac{Q(m)}{K}$.

3.4 Final Protocol

We are now ready to present the protocol desired in Theorem 3. As before, suppose that during iteration t, a message m is chosen to be sent by player i to player j, where m is sampled according to the unknown distribution Q. As defined in Section 3.3, define Q_t^i to be the empirical distribution of the samples seen by player i up to iteration t (which includes the iteration t, where the message seen is m). Similarly define Q_t^j to be the empirical distribution of the samples seen by player j up to iteration t (this includes iteration t, but by definition player j does not see any message in this iteration).

For ease of presentation, we will first assume that the players know the entropy of the distribution Q. This is not a natural assumption, and indeed we do get around it in Section 3.4.2. However, we will describe the main protocol with this assumption to make the analysis more intuitive.

3.4.0.1 Encoding

Upon receiving message m, player i does the following:

- If $t_i < 80 \cdot 2^{2H(Q)/\varepsilon} \cdot \log(8/\varepsilon)$,
  - send the bit $b = 0$
  - use the STATIC protocol (Lemma 4) to send message m.
- Else,
  - send the bit $b = 1$
  - use the UNCERTAIN protocol (Lemma 5) to send message m.

(where the bit b indicates whether player i is using the STATIC protocol or the UNCERTAIN protocol).

3.4.0.2 Decoding

Depending on the value of the received bit b, player j uses either the Static protocol or the Uncertain protocol to decode and output $\widehat{m}$.

3.4.1 Analysis

We now upper-bound the amortized communication cost and the error probability in any iteration of the above protocol.

3.4.1.1 Communication Cost

By the design of the final protocol, each player uses the Static protocol at most $80 \cdot 2^{2H(Q)/\varepsilon} \cdot \log(8/\varepsilon)$ times, and hence overall, the Static protocol is used at most $O(2^{2H(Q)/\varepsilon} \cdot \log(1/\varepsilon) \cdot K)$ times. Thus, if the total number of iterations is T, then the total communication cost in expectation is at most,

$$\underbrace{O\left(2^{2H(Q)/\varepsilon} \cdot \log(1/\varepsilon) \cdot K\right) \cdot (H(Q) + D(Q||P) + 1)}_{\textsc{Static}} + \underbrace{T \cdot O\Big(H(Q) + \log D(Q||P) + O(1)\Big)}_{\textsc{Uncertain}} .$$

And hence, the expected amortized communication cost is at most

$$O\left(H(Q) + \log D(Q||P) + \frac{2^{2H(Q)/\varepsilon} \cdot \log(1/\varepsilon) \cdot K}{T} \cdot D(Q||P) + 1\right).$$

3.4.1.2 Error Guarantee

We first show the following lemma, which is an easy consequence of Markov's inequality.

▶ **Lemma 8.** *For any distribution Q over $\mathbb{N}$, it holds that,*

$$\Pr_{m \sim Q}\left[Q(m) \geq 2^{-H(Q)/\varepsilon}\right] \geq 1 - \varepsilon.$$

Proof. By the definition of the entropy $H(Q)$, we have that $\mathop{\mathbb{E}}_{m \sim Q}\left[\log \frac{1}{Q(m)}\right] = H(Q)$. Thus, the following application of Markov's inequality immediately implies the lemma:

$$\Pr_{m \sim Q}\left[\log \frac{1}{Q(m)} \geq \frac{H(Q)}{\varepsilon}\right] \leq \varepsilon.$$

◀

We will show that in any iteration t, the error probability is at most ε, where the randomness is over all the past and current messages and chosen players. We distinguish two cases:

Case 1. If $t < 8 \cdot 2^{2H(Q)/\varepsilon} \cdot \log(8/\varepsilon) \cdot K$:
Using the Chernoff bound, it is easy to see that

$$\Pr\left[t_i > 80 \cdot 2^{2H(Q)/\varepsilon} \cdot \log(8/\varepsilon) \;\middle|\; t < 2^{2H(Q)/\varepsilon} \cdot \log(8/\varepsilon) \cdot K\right] \leq \exp\left[-\Omega\left(2^{2H(Q)/\varepsilon} \cdot \log(8/\varepsilon)\right)\right] \ll \varepsilon.$$

Thus, it follows that with probability $\geq 1 - \varepsilon$, player i uses the Static protocol in which case there is zero error. Thus, the probability of error is at most ε.

Case 2. If $t \geq 8 \cdot 2^{2H(Q)/\varepsilon} \cdot \log(8/\varepsilon) \cdot K$:
Section 8 implies that when a message m is sampled from Q, with probability at least $1 - \varepsilon/2$ it holds that $Q(m) \geq 2^{-2H(Q)/\varepsilon}$. In this situation, player i may choose to use either the Static or the Uncertain protocol. In the former case, the protocol makes no error. In the latter case, by Lemma 5, the protocol makes error with probability at most

$$2 \cdot e^{-\frac{1}{8}K\frac{t}{Q(m)}} + \frac{\epsilon}{4} \leq 2 \cdot e^{-\frac{1}{8}2^{-2H(Q)/\varepsilon}2^{2H(Q)/\varepsilon}\cdot\log(8/\varepsilon)} + \frac{\epsilon}{4} \ ,$$

which is at most $\varepsilon/2$ if $Q(m) \geq 2^{-2H(Q)/\varepsilon}$. Hence, the total error probability is at most ε.

3.4.2 Getting around the entropy assumption

We let $\varepsilon > 0$ be a sufficiently small positive absolute constant. We now informally describe how to construct a protocol that does not assume that the players know the entropy of the distribution Q. We note that the main reason for the "switching" criterion $t_i < 80 \cdot 2^{2H(Q)/\varepsilon} \cdot \log(8/\varepsilon)$ was to ensure that when we are using the Uncertain protocol and we encounter a message m with $Q(m) \geq 2^{-2H(Q)/\varepsilon}$ (which happens with probability at least $1 - \varepsilon/2$), it holds that $t_i \cdot Q(m) \gg \log(1/\varepsilon)$.

Thus, the protocol guarantees will still hold as long as the players switch to the Uncertain protocol after a sufficiently "large" time t_i. Indeed, we show that it is possible to switch to the Uncertain protocol after time t_i such that $\Pr_{m \sim Q}\left[t_i \cdot Q(m) \gg \log(1/\varepsilon)\right] \geq 1 - \frac{\varepsilon}{4}$.

We now describe the "switching" criterion. In what follows, we prove that for every player, the switching criterion is not met too early, nor is it met too late. Lemma 10 shows that the probability that the switching criterion is met "too early" (i.e., before the time T_0 defined below) is very small. Moreover, it turns out that the probability that the switching criterion is met "too late" (i.e., after time $2^{O\left(\frac{H(Q)}{\epsilon}\right)} \cdot K$) is also very small (see Lemma 9 below). Together, these two properties allow individual players to switch from the Static protocol to the Uncertain protocol based on their observed history of messages. In turn, this allows us to carry out an analysis of the communication cost and the error probability without knowledge of the entropy of Q.

We say that at player i, the switching criterion is met at iteration t_i if

$$t_i \geq \varepsilon^{-3} \quad \text{and} \quad \sum_{m: Q_t^i(m) > t_i^{-\frac{1}{2}}} Q_t^i(m) \geq 1 - \frac{\epsilon}{2}.$$

We first show that, with high probability, the switching criterion is met in time $2^{O\left(\frac{H(Q)}{\epsilon}\right)} \cdot K$

▶ **Lemma 9.** *For every player i, the probability that the switching criterion is met before time $t > 4 \cdot 2^{\frac{16H(Q)}{\epsilon}} K$ is at least $1 - \exp\left(-\frac{1}{64}\epsilon^2 2^{-\frac{4H(Q)}{\epsilon}} \frac{t}{K}\right)$.*

Proof. Let m be such that $Q(m) \geq 2^{-\frac{4H(Q)}{\epsilon}}$. By the Chernoff bound,

$$\Pr\left[Q_t^i(m) \leq (1 - \frac{\epsilon}{4})Q(m)\right] \leq \exp\left(-\frac{1}{32}\epsilon^2 \frac{Q(m)}{K} t\right).$$

Moreover, by the Chernoff bound, we have that $\Pr[t_i \leq \frac{t}{2K}] \leq \exp\left(-\frac{t}{8K}\right)$. We define the event

$$E = \left[t_i \leq \frac{t}{2K} \vee \exists m : Q(m) \geq 2^{-\frac{4H(Q)}{\epsilon}} \wedge Q_t^i(m) \leq (1 - \frac{\epsilon}{4})Q(m)\right].$$

By the union bound, we get that

$$\begin{aligned} \Pr[E] &\leq \exp\left(-\frac{t}{8K}\right) + \exp\left(\frac{4H(Q)}{\epsilon} - \frac{1}{32}\epsilon^2 \frac{2^{-\frac{4H(Q)}{\epsilon}}}{K} t\right) \\ &\leq \exp\left(-\frac{1}{64}\epsilon^2 2^{-\frac{4H(Q)}{\epsilon}} \frac{t}{K}\right) . \end{aligned}$$

If the event E does not hold, then for every m that satisfies $Q(m) > 2^{-\frac{4H(Q)}{\epsilon}}$ we get that

$$Q_t^i(m) > (1 - \frac{\epsilon}{4})Q(m) > (1 - \frac{\epsilon}{4})2^{-\frac{4H(Q)}{\epsilon}} > \left(\frac{t}{2K}\right)^{-\frac{1}{2}} > t_i^{-\frac{1}{2}} .$$

Thus,

$$\begin{aligned} \sum_{m:Q_t^i(m) > t_i^{-\frac{1}{2}}} Q_t^i(m) &\geq \sum_{m:Q(m) > 2^{-\frac{4H(Q)}{\epsilon}}} Q_t^i(m) \\ &\geq (1 - \frac{\epsilon}{4}) \cdot \sum_{m:Q(m) > 2^{-\frac{4H(Q)}{\epsilon}}} Q(m) \\ &= (1 - \frac{\epsilon}{4}) \cdot \Pr_{m \sim Q}\left[Q(m) > 2^{-\frac{4H(Q)}{\epsilon}}\right] \\ &= (1 - \frac{\epsilon}{4}) \cdot \Pr_{m \sim Q}\left[\log \frac{1}{Q(m)} < \frac{4H(Q)}{\epsilon}\right] \\ &\geq (1 - \frac{\epsilon}{4}) \cdot (1 - \frac{\epsilon}{4}) \\ &\geq 1 - \frac{\epsilon}{2}. \end{aligned}$$

Moreover, $t_i > \frac{t}{2K} > \varepsilon^{-3}$. Hence, in this case, the switching criterion is met. ◀

Let T_0 be the smallest $t > \frac{1}{\varepsilon^3 K}$ that satisfies $\Pr_{m \sim Q}\left[Q(m) \geq \frac{1}{4}\sqrt{\frac{K}{t}}\right] > 1 - \frac{3}{4}\epsilon$. First, we will observe that after time T_0, it is indeed safe to switch to the Uncertain protocol.

▶ Observation 3.1. For every time $t \geq T_0$, the Uncertain protocol succeeds with probability at least $1 - \varepsilon$.

Proof. By Lemma 5, with probability at least $1 - \frac{3}{4}\varepsilon$, the protocol succeeds with probability at least $1 - \frac{\varepsilon}{4}$. ◀

It remains to show that with high probability, we will not use the Uncertain protocol before T_0.

▶ **Lemma 10.** *The probability that player i meet the switching criterion before time T_0 is at most ε.*

Proof. We will show that for any fixed $t_i \leq \frac{2T_0}{K}$, we have that the probability that for player i, the switching criterion is met in local time t_i, is at most $2 \cdot \exp\left(-\frac{1}{12}\sqrt{t_i}\right)$. By the union bound, we will get that the probability that for player i, the switching criterion is met before local time $\frac{2T_0}{K}$ is bounded by

$$\sum_{t_i=\epsilon^{-3}+1}^{\infty} 2 \cdot \exp\left(-\frac{1}{12}\sqrt{t_i}\right) \leq \int_{\epsilon^{-3}}^{\infty} 2 \cdot \exp\left(-\frac{1}{12}\sqrt{t_i}\right) dt_i = 24 \cdot (\sqrt{\epsilon^{-3}} + 12) \cdot \exp\left(-\frac{1}{12}\sqrt{\epsilon^{-3}}\right) \leq \frac{\epsilon}{2}.$$

Moreover, by the Chernoff bound, we have that the probability that the local time of player i in (global) time T_0 exceeds $\frac{2T_0}{K}$ is at most $\exp(-\frac{T_0}{3K})$. Thus, the probability that for player i the switching criterion is met before time T_0 is at most $\frac{\varepsilon}{2} + \exp(-\frac{T_0}{3K}) \leq \epsilon$, as required.

Fix t_i and let $M = \left\{m \in \mathbb{N} \mid Q(m) \geq \frac{1}{2\sqrt{t_i}}\right\}$. Since $t_i < \frac{2T_0}{K}$, we have that

$$\Pr_{m\sim Q}[m \in M] = \Pr_{m\sim Q}\left[Q(m) \geq \frac{1}{2\sqrt{t_i}}\right] \leq \Pr_{m\sim Q}\left[Q(m) \geq \frac{\sqrt{K}}{2\sqrt{2T_0}}\right] \leq \Pr_{m\sim Q}\left[Q(m) \geq \frac{\sqrt{K}}{4\sqrt{T_0}}\right] \leq 1 - \frac{3}{4}\epsilon\,.$$

Thus, by the Chernoff bound,

$$\Pr\left[\sum_{m\in M} Q_t^i(m) \geq 1 - \frac{\varepsilon}{2}\right] \leq \exp\left(-\frac{\epsilon \cdot t_i}{25}\right) \tag{1}$$

Now we upper bound $\Pr\left[\exists m \notin M : Q_t^i(m) > \frac{1}{2}\sqrt{\frac{K}{t}}\right]$. To prove this bound, we can assume without lost of generality that for all m except one, we have that $Q(m) > \frac{1}{5\sqrt{t_i}}$: if there exist two elements of such a small probability, we can merge them together to a single element and only increase the probability $\Pr\left[\exists m \notin M : Q_t^i(m) > \frac{1}{2}\sqrt{\frac{K}{t}}\right]$. So we will assume that there are at most $5\sqrt{t_i} + 1$ such elements. By the Chernoff bound, we have that for each $m \notin M$, $\Pr\left[Q_t^i(m) > \frac{1}{\sqrt{t_i}}\right] \leq \exp\left(-\frac{1}{6}\sqrt{t_i}\right)$ and by a union bound we can get that

$$\Pr\left[\exists m \notin M : Q_t^i(m) > \frac{1}{\sqrt{t_i}}\right] \leq (5\sqrt{t_i} + 1) \cdot \exp\left(-\frac{1}{6}\sqrt{t_i}\right) \leq \exp\left(-\frac{1}{12}\sqrt{t_i}\right). \tag{2}$$

By Combining Equations 1 and 2 , assuming $t_i \geq \varepsilon^{-3}$, we get

$$\Pr\left[\sum_{m:Q_t^i(m) > \frac{1}{\sqrt{t_i}}} Q_t^i(m)\right] \leq \Pr\left[\sum_{m\in M} Q_t^i(m) \geq 1 - \frac{\epsilon}{2} \vee \exists m \notin M : Q_t^i(m) > \frac{1}{\sqrt{t_i}}\right] \leq 2 {\cdot} \exp\left(-\frac{1}{12}\sqrt{t_i}\right)\,.$$

This gives an upper bound of $2 \cdot \exp\left(-\frac{1}{12}\sqrt{t_i}\right)$ on the probability that at player i, the switching criterion is met in local time t_i, as needed.

◀

3.5 Efficient Implementation

We briefly sketch how to *efficiently* implement the encoding and decoding strategies of Section 3. The details are deferred to the full version. The overall update time will be linear in $(H(Q) + D(Q||P))/\epsilon$, and the used memory will be proportional to the dictionary-size which is exponential in $(H(Q) + D(Q||P))/\epsilon$. The key question of interest is how to compute the uncertain compression function efficiently. Note that while we would like a fast "processing time" per update, the model naturally allows us to amortize the cost over many operations. In particular, the switch from the Static protocol to the Uncertain one does not have to be carried out in an instant. We will exploit this feature strongly. The corresponding efficient algorithm will have three phases:

1. A phase where we simply use the Static protocol while updating the empirical distributions.
2. A phase where the encoding and decoding dictionaries are being built, but where we still use the Static protocol.
3. A phase where we use the Uncertain protocol.

In what follows, we assume that the messages m and the prior distribution P are presented jointly so that the message m given to player i in round t is $E_P(m)$, namely the Static (Huffman) encoding of m under P. This is a natural assumption about P — after all P is meant to represent a simple and natural, though unoptimized, distribution over the message space. We now recall the statement of Theorem 1.

▶ Theorem 1. Let $\varepsilon > 0$ be a sufficiently small positive absolute constant. For all K and P, there exists a deterministic distributed compression protocol Π, such that for any distribution Q over $\mathbb{N}$ when run for T iterations,

- the amortized communication cost of Π over T iterations approaches $O(H(Q)+\log D(Q||P)+\log(1/\epsilon))$ as T gets large. More formally, the amortized communication cost is
$$O\left(H(Q) + \log D(Q||P) + \log(1/\epsilon) + \frac{2^{\Theta(H(Q)+D(Q||P))/\varepsilon} \cdot K}{T} \cdot D(Q||P) + 1\right)$$
- in each round, the transmitter and receiver run in time linear in their input and output sizes.
- the space usage is exponential in $(H(Q) + D(Q||P))/\epsilon$.

Note that in Theorem 1, the input to the transmitter is $E_P(m)$ and the input to the receiver is the message that she gets from the transmitter.

Proof Sketch. Let $T_\epsilon = 2^{\Theta(H(Q)+D(Q||P))/\epsilon}$ denote the local time at which our inefficient transmitter and receiver – described in the previous section – should switch from the Static protocol to the Uncertain one. In the efficient protocol, during the execution of the Static protocol for the first T_ϵ units of local time, each player will also maintain a count of the number of times she has seen each message using a simple binary tree indexed by $E_P(m)$. At local time T_ϵ, player i updates his empirical distribution $Q^i_{T_\epsilon}$. Note that we can amortize this update time over several rounds. After round T_ϵ, the efficient protocol will start building an encoding and decoding table for the uncertain compression algorithm, but will take $T' = \mathsf{poly}(T_\epsilon)$ rounds to do so (as we will explain below), and in the meanwhile, it will continue using the Static protocol for these T' rounds. At round $T_\epsilon + T'$, it will then switch to the Uncertain protocol, and at this stage it will have a complete table (for all relevant messages) for the encoding and decoding functions, and so it can encode and decode by a simple table lookup.

We also note that the upper bound on the amortized communication cost follows from a similar argument as in the proof of Theorem 3 in Section 3.

So it suffices to show that the encoding and decoding tables can be computed in time $\mathsf{poly}(T_\epsilon)$. A straightforward implementation of the algorithm used in the proof of Theorem 3 essentially works, with a few additional observations. First, we note that we do not need to encode messages m with $P(m) \leq 2^{-\Theta(H(Q)+D(Q||P))/\epsilon}$ since by Markov's inequality such messages occur with probability less than ϵ. This makes sure that the hash families that we need work with a value of N which is at most $2^{(H(Q)+D(Q||P))/\epsilon}$ and the $\log N$ factor in the size of these hash families is equal to $(H(Q) + D(Q||P))/\epsilon$, which is affordable. Next, we use the efficiently computable hash functions which are given by Lemma 7. We apply these hash functions to $E_P(m)$ rather than m in order to make sure that their domain is also small. The upper bound on the encoding time now follows.

For the decoding time, we note that filling in one entry of the decoding table takes time linear in N which is exponentially larger than the budget in the statement of Theorem 1. However, we can divide this task over N rounds while performing $O(1)$ computations per round. The upper bound on the decoding time now follows.

Finally, the space usage is proportional to the size of the encoding and decoding lookup tables which is exponential in $(H(Q) + D(Q||P))/\epsilon$.

◀

3.6 A Computationally Inefficient Protocol with Smaller Communication

In this section, we show that if one does not care about computational efficiency, then we can remove the additive $\log(1/\epsilon)$ term from the communication cost in Theorem 1 while also replacing the multiplicative $2^{\Theta(H(Q)+D(Q||P))/\varepsilon}$ factor by $2^{\Theta(H(Q))/\varepsilon}$. The details are deferred to the full version.

The general structure of the protocol is similar to the one in Section 3 except that for the description and analysis of the UNCERTAIN protocol (Section 3.3). We now describe a computationally inefficient variant of the UNCERTAIN protocol which has smaller communication. The performance of this variant is summarized in the following lemma.

▶ **Lemma 11** (UNCERTAIN Protocol)**.** *Suppose that during iteration t, a message m is chosen to be sent by player i to player j, where m is sampled according to the unknown distribution Q. Then, player i can communicate m to player j, such that the expected communication length is upper bounded by*

$$O\left(H(Q) + \log D(Q||P)\right) + O(1).$$

Morever, the error probability is at most

$$\frac{1}{Q(m)} \cdot \exp\left(-\Omega\left(\frac{t \cdot Q(m)}{K}\right)\right),$$

where the randomness is over all past messages and players chosen in the previous iterations.

We now describe the corresponding encoding and decoding procedures (along with the pre-processing step). Recall Definition 6 of an (N, ℓ, ϵ)-isolating hash family. We now define an (N, ℓ)-isolating hash family to be an $(N, \ell, 1)$-isolating hash family.

3.6.0.1 Pre-Processing Step

As stated earlier, all the players come in with a prior distribution P. In addition, as part of the pre-processing, they compute and store the following:

- Divide the input space $\mathbb{N}$ into a countable number of buckets indexed by $r \in \mathbb{N}_{>0}$, given by $A_r = \left\{m : 2^{-r} < P(m) \leq 2^{-r+1}\right\}$. Clearly, for any r, it holds that $|A_r| \leq 2^r$. In addition, define the function $r(m) := \lceil \log(1/P(m)) \rceil$ for every $m \in \mathbb{N}$, that is, $r(m)$ is the index of the bucket to which m belongs.
- For every r, fix an (arbitrary) choice of isolating hash families $\mathcal{H}_{(N,\ell)}$, for $N = |A_r|$ and every choice of $\ell \in \{1, 2, \cdots, \lceil \log N \rceil\}$.

Suppose during iteration t, a message m is chosen to be sent by player i to player j, where m is sampled according to the unknown distribution Q. Define Q_t^i to be the empirical distribution of the samples seen by player i up to iteration t (which includes the iteration t, where the message seen is m). Similarly, define Q_t^j to be the empirical distribution of the samples seen by player j up to iteration t (this includes iteration t, but by definition player j does not see any message in this iteration). The players use the encoding and decoding strategies described next.

3.6.0.2 Encoding

Upon receiving message m, player i does the following,

(i) let $A \overset{\text{def}}{=} A_{r(m)}$ and $N \overset{\text{def}}{=} |A|$.
(ii) let $S \overset{\text{def}}{=} \left\{m' \in A \setminus \{m\} : Q_t^i(m') \geq \frac{1}{16} Q_t^i(m)\right\}$.
(iii) let $\ell = \lceil \log |S| \rceil$.
(iv) let $u \in [|\mathcal{H}_{(N,\ell)}|]$ and $h_u \in \mathcal{H}_{(N,\ell)}$ such that $h_u(m) \notin h_u(S)$.
(v) Send the tuple $(r, \ell, u, h_u(m))$ to player j.

Note that the property of isolating hash families (see Definition 6) guarantees the existence of $h_u \in \mathcal{H}_{(N,\ell)}$ as desired in (iv).

The intuition for this encoding is as follows: upon receiving r, player j understands that $m \in A_r$, upon receiving ℓ, she understands which hash family to use, upon receiving u, she knows which hash function to use, and hopefully with $h_u(m)$, she will be able to recover m correctly.

3.6.0.3 Decoding

Upon receiving the tuple (r, ℓ, u, h_*), player j does the following:

(i) Set $A = A_r$ and $N \overset{\text{def}}{=} |A|$.
(ii) Identify $h_u \in \mathcal{H}_{(N,\ell)}$.
(iii) Output $\arg\max_{m' \in A : h_u(m') = h_*} Q_t^j(m')$.

The analysis of the communication cost and the error guarantee appears in Appendix A, where Lemma 11 is proved.

References

1 Douglas Adams. *The Hitchhiker's Guide to the Galaxy #1.* Del Rey, 1979.

2 Noam Chomsky. Reflections on language. *New York*, 3, 1975.

3 Noam Chomsky. Rules and representations. *Behavioral and brain sciences*, 3(01):1–15, 1980.

4 Elad Haramaty and Madhu Sudan. Deterministic compression with uncertain priors. In *Proceedings of the 5th conference on Innovations in theoretical computer science*, pages 377–386. ACM, 2014.

5 Marc D Hauser, Noam Chomsky, and W Tecumseh Fitch. The faculty of language: what is it, who has it, and how did it evolve? *science*, 298(5598):1569–1579, 2002.

6 Brendan Juba, Adam Tauman Kalai, Sanjeev Khanna, and Madhu Sudan. Compression without a common prior: an information-theoretic justification for ambiguity in language. In *Innovations in Computer Science - ICS*, pages 79–86. Tsinghua University Press, 2011.

7 Simon Kirby, Mike Dowman, and Thomas L Griffiths. Innateness and culture in the evolution of language. *Proceedings of the National Academy of Sciences*, 104(12):5241–5245, 2007.

8 Erez Lieberman, Jean-Baptiste Michel, Joe Jackson, Tina Tang, and Martin A Nowak. Quantifying the evolutionary dynamics of language. *Nature*, 449(7163):713–716, 2007.

9 Martin A Nowak. Evolutionary biology of language. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 355(1403):1615–1622, 2000.

10 Martin A Nowak and Natalia L Komarova. Towards an evolutionary theory of language. *Trends in cognitive sciences*, 5(7):288–295, 2001.

11 Martin A Nowak, Natalia L Komarova, and Partha Niyogi. Computational and evolutionary aspects of language. *Nature*, 417(6889):611–617, 2002.

12 Martin A Nowak and David C Krakauer. The evolution of language. *Proceedings of the National Academy of Sciences*, 96(14):8028–8033, 1999.

13 Martin A Nowak, Joshua B Plotkin, and David C Krakauer. The evolutionary language game. *Journal of Theoretical Biology*, 200(2):147–162, 1999.

14 Steven Pinker and Paul Bloom. Natural language and natural selection. *Behavioral and brain sciences*, 13(04):707–727, 1990.

15 Joshua B Plotkin and Martin A Nowak. Language evolution and information theory. *Journal of Theoretical Biology*, 205(1):147–159, 2000.

16 Giacomo Rizzolatti and Michael A Arbib. Language within our grasp. *Trends in neurosciences*, 21(5):188–194, 1998.

A Analysis of Computationally Inefficient Protocol

We now analyze the operation of the protocol described in Section 3.6.

A.0.0.1 Communication Cost

Suppose the message m is chosen to be sent by player i to player j. The communication cost of sending the tuple $(r, \ell, u, h_u(m))$ is as follows:

(i) $\log\lceil\log(1/P(m))\rceil$ bits to send r.
(ii) $\log(\log(1/Q^i_t(m))+5)$ bits to send ℓ, since $\ell \le \log|S|+1 \le \log(16/Q^i_t(m))+1$.
(iii) $\log(1/Q^i_t(m)) + \log\lceil\log(1/P(m))\rceil + 5$ bits to send u (it takes $\ell + \log N$ bits).
(iv) $\log(1/Q^i_t(m)) + 5$ bits to send $h_u(m)$.

Thus, the total communication is given by,

$$\underbrace{2\log(1/Q^i_t(m))}_{(I)} + \underbrace{2\log\lceil\log(1/P(m))\rceil}_{(II)} + \underbrace{\log(\log(1/Q^i_t(m))+5)+10}_{(III)}$$

We wish to prove guarantees on the expected communication cost, when m is drawn from Q. The terms in (III) are lesser order terms, which are smaller than (I), thus we choose to ignore them. Term (II) in expectation is,

$$\mathop{\mathbb{E}}_{m\sim Q}\left[\log\left(\left\lceil\log\frac{1}{P(m)}\right\rceil\right)\right] \quad\le\quad \log\left(\mathop{\mathbb{E}}_{m\sim Q}\left\lceil\log\frac{1}{P(m)}\right\rceil\right) \quad\le\quad \log(H(Q)+D(Q||P)+1)$$

Term (I) is slightly more tricky to bound in expectation. Note that the empirical distribution changes on receiving message m (this turns out to be critical in bounding the communication!). That is, $Q^i_t(m) = \frac{1+(t-1)Q^i_{(t-1)}(m)}{t}$. Also let $\mathcal{M}^i_t$ be the multi-set of all messages that player i has seen up to time t. Thus, Term (I) in expectation is as follows,

$$\mathop{\mathbb{E}}_{\mathcal{M}^i_{(t-1)}}\mathop{\mathbb{E}}_{m\sim Q}\left[\log\frac{1}{Q^i_t(m)}\right] \quad=\quad H(Q) + \mathop{\mathbb{E}}_{\mathcal{M}^i_{(t-1)}}\mathop{\mathbb{E}}_{m\sim Q}\left[\log\frac{Q(m)}{\frac{1}{t_i}+\frac{(t_i-1)Q^i_{(t-1)}(m)}{t_i}}\right]$$

In order to bound the second term above, we consider two cases, (i) $Q^i_{(t-1)}(m) \ge Q(m)/2$ or (ii) $Q^i_{(t-1)}(m) < Q(m)/2$. After fixing t_i and m, by Chernoff bound over the randomness of $\mathcal{M}^i_{(t-1)}$ we have that case (i) happens with probability at least $1-\exp(-t\cdot Q(m)/8)$.

$$\text{Case (i) } Q^i_{(t-1)}(m) \ge Q(m)/2 \quad\Longrightarrow\quad \log\left(\frac{Q(m)}{\frac{1}{t_i}+\frac{(t-1)Q^i_{(t)}(m)}{t_i}}\right) \le 1$$

$$\text{Case (ii) } Q^i_{(t-1)}(m) < Q(m)/2 \quad\Longrightarrow\quad \log\left(\frac{Q(m)}{\frac{1}{t_i}+\frac{(t_i-1)Q^i_{(t-1)}(m)}{t_i}}\right) \le \log(t_i\cdot Q(m))$$

Using these upper bounds we get that,

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathcal{M}^i_{(t-1)}} \mathop{\mathbb{E}}_{m\sim Q}\left[\log \frac{Q(m)}{\frac{1}{t}+\frac{(t_i-1)Q^i_{(t-1)}(m)}{t_i}}\right] &\le \mathop{\mathbb{E}}_{m\sim Q}\left[1\cdot\left(1-e^{-t_i\cdot Q(m)/8}\right)+\log(t_i\cdot Q(m))\cdot e^{-t_i\cdot Q(m)/8}\right]\\
&\le 1+\mathop{\mathbb{E}}_{m\sim Q}\left[\log(t_i\cdot Q(m))\cdot e^{-t_i\cdot Q(m)/8}\right]\\
&\le 2\,,
\end{aligned}$$

where the last inequality just follows from the fact that $\log(x)\cdot e^{-x/8}\le 1$ for all x.

Thus the overall communication is bounded by

$$(2+o(1))\,H(Q)+2\log D(Q||P)+O(1)\,.$$

A.0.0.2 Error Guarantee

We now show that the error probability in iteration t, denoted by p^{err}_t of the protocol is upper bounded by $\frac{1}{Q(m)}\cdot 2^{-\Omega\left(\frac{t\cdot Q(m)}{K}\right)}$, where m is fixed to be the message sent in round t.

We first give an intuitive explanation for the error bound. Since player i has communicated (r,ℓ,u), player j knows the correct bucket of messages A_r to which m belongs. Knowing ℓ and u, player j also knows which hash function is being used, which is chosen to ensure that for every $m'\in S\setminus\{m\}$, $h_u(m)\neq h_u(m')$. Thus, the only way in which an error can happen is that there exists some $m'\notin S$ such that $h_u(m)=h_u(m')$ and $Q^j_t(m')>Q^j_t(m)$.

Since $m'\notin S$, it implies by definition of S that $Q^i_t(m')\le Q^i_t(m)/16$, which means that player i has seen the message m' significantly fewer times compared to the message m. On the other hand, we also have that $Q^j_t(m')>Q^j_t(m)$, which means that player j has seen the message m' at least as many times as message m. For "large" t, it is very unlikely that players i and j have seen m and m' in such disproportionate manner.

To make the arguments go through, we need to union bound over all $m'\in A_r\setminus S$. However, a naive union bound is too lossy because we do not have any reasonable upper bound on the number of m's. To get around this issue, we do a simple bucketing argument.

The formal upper bound on p_t^{err} is shown as follows,

$$\begin{aligned}
p_t^{\text{err}} &= \Pr[\exists m' \in A:\ h_u(m') = h_u(m) \text{ and } Q_t^j(m') > Q_t^j(m)] \\
&\le \Pr\left[\exists m' \in A:\ Q_t^i(m') < \frac{1}{16}Q_t^i(m) \text{ and } Q_t^j(m') > Q_t^j(m)\right] \\
&\le \Pr\left[\exists m' \in A:\ Q(m') > \frac{1}{4}Q(m) \text{ and } Q_t^i(m') < \frac{1}{16}Q_t^i(m)\right] \\
&\quad + \Pr\left[\exists m' \in A:\ Q(m') \le \frac{1}{4}Q(m) \text{ and } Q_t^j(m') > Q_t^j(m)\right] \\
&\le \Pr\left[\exists m' \in A:\ Q(m') > \frac{1}{4}Q(m) \text{ and } Q_{t-1}^i(m') < \frac{1}{16}\left(Q_{t-1}^i(m) + \frac{1}{t_i - 1}\right)\right] \\
&\quad + \Pr\left[\exists m' \in A:\ Q(m') \le \frac{1}{4}Q(m) \text{ and } Q_t^j(m') > Q_t^j(m)\right] \\
&\le \underbrace{\Pr\left[\exists m' \in A:\ Q(m') > \frac{1}{4}Q(m) \text{ and } Q_{t-1}^i(m') < \frac{1}{8}Q(m)\right]}_{(I)} \\
&\quad + \underbrace{\Pr\left[Q_{t-1}^i(m) + \frac{1}{t_i - 1} > 2 \cdot Q(m) \;\middle|\; t_i \ge \frac{t}{2K}\right]}_{(II)} + \underbrace{\Pr\left[t_i \le \frac{t}{2K}\right]}_{(III)} \\
&\quad + \underbrace{\Pr\left[\exists m' \in A:\ Q(m') \le \frac{1}{4}Q(m) \text{ and } Q_t^j(m') > \frac{1}{2}Q(m)\right]}_{(IV)} \\
&\quad + \underbrace{\Pr\left[Q_t^j(m) < \frac{1}{2}Q(m)\right]}_{(V)} .
\end{aligned}$$

We bound each term individually. Firstly, since $\left\{Q_{t-1}^i(m)|t_i\right\}$ (i.e., $Q_{t-1}^i(m)$ conditioned on a fixed t_i), t_i and $Q_t^j(m)$ are binomial random variables with probabilities $Q(m)$, $\frac{1}{K}$ and $\frac{Q(m)}{K}$ respectively, the terms (II), (III) and (V) are easily upper bounded using the Chernoff bound. In particular,

$$\begin{aligned}
\Pr\left[Q_{t-1}^i(m) + \frac{1}{t_i - 1} > 2 \cdot Q(m) \;\middle|\; t_i \ge \frac{t}{2K}\right] &\le \exp\left(-\Omega\left(\frac{t \cdot Q(m)}{K}\right)\right) \\
\Pr\left[t_i \le \frac{t}{2K}\right] &\le \exp\left(-\Omega\left(\frac{t \cdot Q(m)}{K}\right)\right) \\
\Pr\left[Q_t^j(m) < \frac{1}{2}Q(m)\right] &\le \exp\left(-\Omega\left(\frac{t \cdot Q(m)}{K}\right)\right) .
\end{aligned}$$

Term (I) is also upper bounded by Chernoff bound and a union bound over m', since the number of m' satisfying $Q(m') > \frac{1}{4}Q(m)$ is at most $4/Q(m)$. Thus,

$$\Pr\left[\exists m' \in A:\ Q(m') > \frac{1}{4}Q(m) \text{ and } Q_t^i(m') < \frac{1}{8}Q(m)\right] \le \frac{4}{Q(m)} \cdot \exp(-\Omega(t_i \cdot Q(m))).$$

To bound term (IV), we can assume without loss of generality that there is at most one $m' \in A$, such that, $Q(m') \le \frac{1}{8}Q(m)$. This is because, if there were to exist $m_1', m_2' \in A$, such that, $Q(m_1'), Q(m_2') \le \frac{1}{8}Q(m)$, then we can identify m_1' and m_2' as the same message m_0'. Note that we can do this because we will still have that $\Pr[Q(m_0') \le \frac{1}{4}Q(m)]$ and

$$\Pr\left[Q_t^j(m_1') > \frac{1}{2}Q(m) \text{ or } Q_t^j(m_2') > \frac{1}{2}Q(m)\right] \le \Pr\left[Q_t^j(m_0') > \frac{1}{2}Q(m)\right]$$

Thus, to bound term (IV), we can again use a Chernoff bound and a union bound over m', since the number of m' such that $Q(m') > \frac{1}{8}Q(m)$ is at most $8/Q(m)$. Thus, we get that,

$$\Pr\left[\exists m' \in A: \ Q(m') \le \frac{1}{4}Q(m) \text{ and } Q_t^j(m') > \frac{1}{2}Q(m)\right] \le \frac{1}{Q(m)} \cdot \exp\left(-\Omega\left(\frac{t \cdot Q(m)}{K}\right)\right)$$

Thus, overall in any individual round, we have that,

$$p_t^{\text{err}} \le \frac{1}{Q(m)} \cdot \exp\left(-\Omega\left(\frac{t \cdot Q(m)}{K}\right)\right).$$

This concludes the proof of Lemma 5.