The Compression and Concentration of Classical and Quantum Information

Andrew Ross[1,*] and Peter Love[1]

[1]Physics Department, Haverford College, Haverford, PA 19041

May 5, 2011

Abstract

In this paper we discuss classical and quantum information theory with a focus on data compression and entanglement concentration. We attempt to define the information-theoretic properties of physical systems and quantify the amount of information needed to represent them. We begin by overviewing basic results about independent and identically distributed (i.i.d.) sources before moving on to discuss non-i.i.d. grand canonical ensembles of fermions and bosons and states of entangled qubits. We present algorithms to compress bits and qubits from i.i.d. sources, concentrate entanglement between two distant qubits, and compress data from ensembles of fermions and bosons. We consider the limitations on entanglement concentration algorithms for three qubits.

1 Introduction

All information is stored in physical systems. Even a basic unit of information such as a bit, which describes a digit that is 0 or 1, must always be embodied in some sort of system that has two states. In most computers a bit is implemented with a circuit that can have two distinct voltages; in other settings bits are implemented by the direction of a light switch, the presence of a flag on a ship, or a simple "yes" or "no." Each of these systems is quite different, but we can represent the abstract mathematical concept of a bit in all of them.

There are other physical systems that store or output information, but we often need concepts more complicated than simple classical bits to describe them. In this paper we consider a number of different "message sources," which in later sections will closely correspond to a number of different physical systems, that output information in discrete blocks we call "messages." This conceptual framework for describing sources of information is applicable to a wide range

*To whom correspondence should be addressed. Email: aross@haverford.edu

of systems. A central focus of the paper will be on quantifying the information contained in messages or sequences of messages from particular message sources, and on constructing algorithms to move or concentrate the information contained in messages into as few physical systems as possible.

1.1 Classical Information Theory

In classical information theory, we model a message source as something that outputs a sequence of random variables, say $X_1, X_2, \cdots, X_n$. These variables take values in an alphabet of symbols, which can be finite or infinite. We will only consider finite alphabets (e.g. the english alphabet, and not the set of integers).

1.1.1 An i.i.d. Message Source

We make two more restrictions on the definition of our source. First, measurements of variables must be independent. That is, the probability of measuring X_1 to be some particular letter of the alphabet must not be dependent on the history of past measurements. Mathematically, that means that the total probability of measuring a sequence $X_1, X_2, \cdots, X_n$ should be equal to the product of $P(X_1), P(X_2), \cdots$, and $P(X_n)$. Second, variables must be identically distributed. Identically distributed does *not* mean that, when we measure X_1, it must have an equal probability of being any of the letters in the alphabet from which it draws its values. Rather it means that the probability distribution of values for X_1 must be equal to the probability distribution of values for X_2. For example, imagine you have N coins and you decide to create a string of N random variables by flipping the first coin, then the second one, and so on. This N coin message source is independent if the results of each coin flip do not depend on the results of previous coin flips. It is identically distributed if each coin has the same probability distribution of heads and tails. Sources which are both independent and identically distributed are abbreviated as i.i.d. For many sources (such as repeated flips of the same coin), it is difficult to separate the concepts of independence and identical distribution because they often seem to overlap, but hopefully this example elucidates the distinction.

1.1.2 Type Classes

Assume we have an i.i.d. source whose alphabet has two letters. We can represent its output as a string of bits by assigning one letter to "0" and the other to "1." The type of such a string, also known as its Hamming weight, is the number of entries which are "1." We define a type class (N, T) as the set of all bit strings of length N with type T.

For i.i.d. sources, all strings in a type class have an equal probability of being drawn. This is because the probability of drawing a particular string from an i.i.d. source is invariant under permutations on its elements. The probability of drawing "01," for example, is equal to that of drawing "10" because sampling

each letter is an independent event, and the probability distribution from which we acquire them is identical for both drawings. Since type classes are closed under permutations, all strings in a type class have equal probability.

The number of strings in a type class is given by the binomial coefficient

$$\binom{N}{T} = \frac{N!}{T!(N-T)!}. \tag{1}$$

We define $\binom{p}{0} \equiv 1$ for any $p > 0$ and $\binom{q}{r} \equiv 0$ when $r > q$. Binomial coefficients obey a simple recursion rule

$$\binom{N}{T} = \binom{N-1}{T} + \binom{N-1}{T-1}, \tag{2}$$

which may be diagrammed in a triangular array known as Pascal's triangle, shown in Figure 1. The equal probability of i.i.d.-sourced strings within a par-

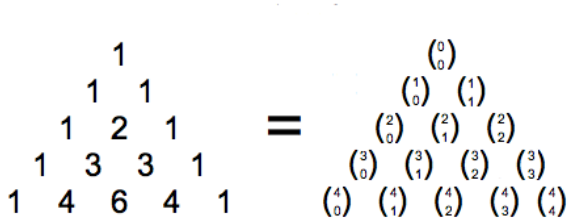

Figure 1: Pascal's Triangle. Defining the top entry of the triangle to be $\binom{0}{0} \equiv 1$, we find the value at any other entry by summing up the values of the two entries above it, except at the edges, which are all set to 1. The result is a triangular array of the binomial coefficients, where the value at the Tth element of the Nth row is equal to $\binom{N}{T}$.

ticular type class will be critical to some of the algorithms we introduce later, and Pascal's triangle provides us with a convenient graphical way of thinking about them.

1.1.3 Shannon Entropy

The information content of an i.i.d. message source X is quantified by the Shannon entropy

$$H(X) = -\sum_{a} p(a) \log_2 p(a), \tag{3}$$

where the sum is over all messages in the alphabet and the base two logarithm indicates that we are measuring in bits [10]. For the remainder of this paper, $\log(x)$ will denote the base two logarithm of x.

For a message source emitting fair coin flips, or any source with $p(0) = p(1) = \frac{1}{2}$, the Shannon entropy is simply

$$-\frac{1}{2}\log\left(\frac{1}{2}\right) - \frac{1}{2}\log\left(\frac{1}{2}\right) = \log(2) = 1 \text{ bit}. \tag{4}$$

We need one bit to represent two values, so we might expect any two-outcome source to output one bit of information. However, for a message source emitting unfair coin flips, say heads 70% of the time and tails 30% of the time,

$$H = -\frac{7}{10}\log\left(\frac{7}{10}\right) - \frac{3}{10}\log\left(\frac{3}{10}\right) = 0.88 \text{ bits}. \tag{5}$$

How do we interpret this? We cannot physically represent 0.88 bits on a classical computer and would need to use a full physical bit to store the result of the unfair coin flip. However, Shannon's noiseless channel coding theorem [10] states that we only need $nH(X)$ bits to represent a sequence of n variables from i.i.d. source X as $n \to \infty$. The low entropy of an unfair coin flip is relevant not for single outputs from the source but for long sequences of outputs.

The proof of Shannon's theorem centers on the idea of *typical sequences* of outputs. As $n \to \infty$, there is a vanishing probability of drawing any length-n sequence not in a set of typical sequences. For an i.i.d. source emitting 0 with probability p and 1 with probability $1-p$, the typical set is the set of all length-n sequences with *type* $n(1-p)$, i.e. with np 0s and $n(1-p)$ 1s. Assume we measure n random variables from this source and obtain the sequence $x_1, x_2, \cdots, x_n$. The probability of obtaining this sequence in particular is

$$p(x_1, x_2, \cdots, x_n) = p(x_1)p(x_2)\cdots p(x_n) \approx p^{np}(1-p)^{n(1-p)}, \tag{6}$$

where we have used the fact that X is independent to convert $p(x_1, x_2, \cdots, x_n)$ to a product of individual probabilities, the fact that X is identically distributed to conclude that each of those probabilities is either p or $1-p$, and the fact that we have almost certainly obtained a typical sequence to conclude that np of those probabilities are p and $n(1-p)$ of them are $1-p$.

All typical sequences are equally likely because they are members of the same type class, so $p(x_1, x_2, \cdots, x_n)$ is not simply the probability of obtaining a particular sequence of variables but the probability of obtaining any typical sequence. Therefore the number of typical sequences is $\frac{1}{p(x_1, x_2, \cdots, x_n)}$ or $p^{-np}(1-p)^{-n(1-p)}$. Taking the logarithm-base-two of this expression, we see that

$$\log\left(p^{-np}(1-p)^{-n(1-p)}\right) = -np\log(p) - n(1-p)\log(1-p),$$

which is just $nH(X)$, the entropy of the source times the length of the sequence. Therefore there are $2^{nH(X)}$ likely sequences.

We only ever need $\log(N)$ bits to count N things, and to count N things is to uniquely specify each one of them. If we obtain a random output $X_1, X_2, \cdots, X_n$

from X, we know it is one of $2^{nH(X)}$ likely sequences. Since we only need $nH(X)$ bits to uniquely specify which likely sequence it is, we really only need $nH(X)$ bits to store it, not n. This is the essence of Shannon's noiseless coding theorem, which is a foundational result in classical information theory [10].

We can think of the Shannon entropy as the amount of information we gain when we measure a random variable, or, equivalently, our uncertainty about the value of a random variable before we measure it [7]. For example, a fair coin gives us one bit of new information every time we flip it, and we only flip it because we are completely uncertain of its outcome beforehand. More concretely, the Shannon entropy is the minimum number of bits, divided by n, that we need to store all information contained in a string of random variables of length n. It is only because the Shannon entropy quantifies a *lower bound* for the resources we need to store all the information in a sequence that it has the qualitative meaning of describing how much information a sequence actually contains, and how much we learn by measuring it.

Note how the concept of data compression has surfaced in our discussion of entropy. If a string of length n only contains $nH(X)$ bits of information, then why not store it in that many bits? The idea of likely sequences immediately suggests a scheme for compression: order the set of all sequences by their relative likelihood. Map the most likely sequence to 0 (binary 0), the next most likely as 1 (binary 1), the next most likely as 10 (binary 2), and so on. Sequences with the same probability can be ordered in any way desired. Stop when you have labeled $nH(X)$ sequences. Replace your string of length n with its shorter label. If you can reconstruct the ordering of states based on your knowledge of the message source, you can recreate the original string from its label. This is a basic algorithm for data compression that follows naturally from Shannon's theorem, and we will discuss a particular implementation of it in later sections.

1.2 Quantum Information Theory

Just as we can have a classical message source that outputs *variables* from an alphabet, we can also have a quantum message source that outputs physical quantum systems occupying *states* drawn from a finite or infinite set of possible states. In this section we explore how to mathematically describe quantum information sources, how we can adapt classical concepts like type to their more complex mathematical structure, and how we can quantify the information contained in what they output.

1.2.1 Quantum i.i.d. Message Source

In quantum information theory, a message source outputs quantum systems occupying states that we represent as vectors in a complex, N-dimensional Hilbert space. We usually pick a particular set of orthogonal basis vectors labelled by bitstrings $\{|0\rangle, |1\rangle, \cdots |N-1\rangle\}$ for this Hilbert space, which we call the computational basis states, and write our outputs as superpositions of them. Restricting ourselves to the 2-dimensional case, a quantum message source outputs

qubits

$$|\psi_i\rangle = a_i |0\rangle + b_i |1\rangle \tag{7}$$

from an ensemble

$$\{p_i, |\psi_i\rangle\}.$$

Here a_i and b_i are complex probability amplitudes defining each $|\psi_i\rangle$, while p_i are classical probabilities indicating a lack of classical knowledge about which $|\psi_i\rangle$ the source emits. For an i.i.d. quantum message source, we pick $|\psi_i\rangle$s from the ensemble with the same i.i.d. restriction as in the classical case: $p_i, |\psi_i\rangle$ is exactly the same for every sampling event regardless of what we have drawn before.

A quantum i.i.d. source is fully characterized by a density matrix

$$\rho = \sum_i p_i |\psi_i\rangle \langle\psi_i|. \tag{8}$$

ρ is always Hermitian and has positive eigenvalues. The diagonal terms of the density matrix ($|0\rangle \langle 0|$ and $|1\rangle \langle 1|$) represent the probabilities of obtaining the computational basis states when we measure the output of source ρ in that basis. If we apply a change of basis from $\{|0\rangle, |1\rangle\}$ to $\{|0'\rangle, |1'\rangle\}$, then the diagonals of the transformed ρ' represent the probabilities of measuring $|0'\rangle$ and $|1'\rangle$. $\mathrm{Tr}(\rho)$, the sum of the diagonal terms, equals one in every basis. The off-diagonal terms ($|0\rangle \langle 1|$ and $|1\rangle \langle 0|$), if nonzero, indicate that the source is emitting states that are superpositions of basis states. These states are not orthogonal to the basis states and thus cannot be reliably distinguished from them by any single measurement.

We can obtain the density matrix of a single known state $|\psi\rangle$, also known as a pure state, by constructing the projector $|\psi\rangle \langle\psi|$. The density matrix of a pure state has the special property that

$$\mathrm{Tr}(\rho^2) = 1. \tag{9}$$

If, on the other hand, $\mathrm{Tr}(\rho^2) < 1$, then ρ represents a quantum system about which we have classical ignorance. In general we say that if $\mathrm{Tr}(\rho^2) \leq 1$, then ρ represents a mixed state. We can think of a quantum i.i.d. message source either as something that emits different states from an ensemble with i.i.d. restrictions, or as a source of identical mixed states ρ.

1.2.2 Quantum Types

The classical notion of types is inherently basis-dependent. It depends on the existence of set of perfectly distinguishable symbols {0,1} that our message source outputs. Quantum message sources, however, may output states that are superpositions of computational basis states $|0\rangle$ and $|1\rangle$. We could attempt a naive approach towards quantum types and define the type of a string of qubits as the number of $|1\rangle$ outcomes when we measure it in the $\{|0\rangle, |1\rangle\}$ basis, but this definition will not be useful.

For example, consider the quantum message source

$$\rho = \begin{pmatrix} \frac{1}{2} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{2} \end{pmatrix}. \tag{10}$$

We draw four qubits from this source to obtain a string

$$s = |\psi_1\rangle |\psi_2\rangle |\psi_3\rangle |\psi_4\rangle . \tag{11}$$

We could measure each qubit of s in the $\{|0\rangle, |1\rangle\}$ basis and call the number of $|1\rangle$ outcomes the type T of s. The diagonal terms of the density matrix tell us that $p(|0\rangle) = p(|1\rangle) = \frac{1}{2}$, so on average we should draw strings for which we measure an equal proportion of $|0\rangle$s and $|1\rangle$s. However, the off-diagonal terms of the density matrix are non-zero, indicating that $|\psi_i\rangle$ may be superpositions of the basis states.

In measuring the type, then, we must measure superpositions, which causes them to irreversibly collapse to either $|0\rangle$ or $|1\rangle$. We lose our original string in the process of measuring its type. Furthermore, if we had multiple copies of s, we would not measure the same type for all of them. And if we measured in a different basis ($\{|0\rangle, |1\rangle\}$ was an arbitrary choice) we would get a different probability distribution of possible type values. This is not a satisfying way of measuring or defining the type of a string of qubits.

However, we can always find some unitary matrix U such that

$$\rho = U \begin{pmatrix} \lambda_0 & 0 \\ 0 & \lambda_1 \end{pmatrix} U^\dagger. \tag{12}$$

The columns of U, which by its unitarity are orthogonal, define a basis $\{|\lambda_0\rangle, |\lambda_1\rangle\}$. In this basis our message source only outputs $|\lambda_0\rangle$ with probability λ_0 and $|\lambda_1\rangle$ with probability λ_1. From here we can meaningfully calculate something corresponding to a classical type by counting the number of qubits which are $|\lambda_1\rangle$.

Returning to our example, we find that we can diagonalize ρ if we express it in the $\{|+\rangle, |-\rangle\}$ basis, where we have defined orthogonal states

$$|+\rangle \equiv \frac{|0\rangle + |1\rangle}{\sqrt{2}} \tag{13}$$

and

$$|-\rangle \equiv \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{14}$$

Using the unitary Hadamard matrix

$$H \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{15}$$

whose columns are $|+\rangle$ and $|-\rangle$ expressed in the $\{|0\rangle, |1\rangle\}$ basis, we can map $|0\rangle \to |+\rangle$ and $|1\rangle \to |-\rangle$, which corresponds to a rotation of our basis vectors.

Thus we can transform

$$\begin{aligned}\rho &\rightarrow H\rho H^\dagger \\ &= \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} \frac{1}{2} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{2} \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\ &= \begin{pmatrix} \frac{2}{3} & 0 \\ 0 & \frac{1}{3} \end{pmatrix}. \end{aligned} \tag{16}$$

In other words, we can interpret ρ as describing an ensemble

$$\frac{2}{3}\left|+\right\rangle\left\langle+\right| + \frac{1}{3}\left|-\right\rangle\left\langle-\right|,$$

and it is clear that measuring states drawn from this ensemble in the $\{\left|+\right\rangle, \left|-\right\rangle\}$ basis will simply give us type information about s without changing it.

This does not mean we have to interpret $\left|\psi_i\right\rangle$ as truly being $\left|+\right\rangle$ or $\left|-\right\rangle$. We have shown that ρ equals $\frac{2}{3}\left|+\right\rangle\left\langle+\right| + \frac{1}{3}\left|-\right\rangle\left\langle-\right|$. However, it is easy to show that ρ also equals

$$\frac{1}{3}\left|0\right\rangle\left\langle0\right| + \frac{1}{3}\left|1\right\rangle\left\langle1\right| + \frac{1}{3}\left|+\right\rangle\left\langle+\right|,$$

so we could just as easily interpret all of the qubits as $\left|0\right\rangle$, $\left|1\right\rangle$, or $\left|+\right\rangle$. Because a density matrix fully describes a message source, both of these ensembles have the exact same statistical properties. For either ensemble, if we measure in $\{\left|0\right\rangle$, $\left|1\right\rangle\}$, we obtain approximately half $\left|0\right\rangle$ and half $\left|1\right\rangle$, and if we measure in $\{\left|+\right\rangle$, $\left|-\right\rangle\}$, we obtain approximately $\frac{2}{3}\ \left|+\right\rangle$ and $\frac{1}{3}\ \left|-\right\rangle$. It does not matter how we interpret the individual qubits because it is impossible to distinguish the two ensembles. This is known as the unitary freedom of density matrices.

Because the density matrix is not diagonal in $\{\left|0\right\rangle, \left|1\right\rangle\}$, measuring the state of our string collapses it to a string of $\left|0\right\rangle$s and $\left|1\right\rangle$s. If we defined a new message source by drawing qubits at random from this collapsed string, its density matrix would be

$$\frac{I}{2} = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}.$$

If we measured qubits from this new source in the $\{\left|+\right\rangle, \left|-\right\rangle\}$ basis, we would not find $\frac{2}{3}\ \left|+\right\rangle$ and $\frac{1}{3}\ \left|-\right\rangle$ but an equal proportion of them, since

$$\frac{I}{2} = \frac{1}{2}\left|+\right\rangle\left\langle+\right| + \frac{1}{2}\left|-\right\rangle\left\langle-\right|.$$

Measuring our original string in the $\{\left|0\right\rangle, \left|1\right\rangle\}$ basis changes the statistical properties of that string in other bases.

However, if we measure in the $\{\left|+\right\rangle, \left|-\right\rangle\}$ basis, our original density matrix

$$\rho = \begin{pmatrix} \frac{1}{2} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{2} \end{pmatrix}$$

will continue to describe the statistical properties of the string, even if we change basis. It's not as though $\frac{2}{3}\left|+\right\rangle\left\langle+\right| + \frac{1}{3}\left|-\right\rangle\left\langle-\right|$ is an especially privileged ensemble,

or that we can say that $|\psi_i\rangle$ really *are* $|+\rangle$ and $|-\rangle$, but rather that the $\{|+\rangle, |-\rangle\}$ basis is the only one in which measurement makes no discernible change to s. It is only in the diagonal basis $\{|\lambda_0\rangle, |\lambda_1\rangle\}$ of ρ that we can measure something analogous to type without changing any measurable property of the system. Thus it is convenient for us to choose to interpret the density matrix as describing the ensemble

$$\lambda_1 |\lambda_1\rangle \langle\lambda_1| + \lambda_2 |\lambda_2\rangle \langle\lambda_2|, \tag{17}$$

and we have the unitary freedom to do so.

Note that in this quantum case, we needed *two* variables to effectively describe the type of our state: T, the number of entries which were $|\lambda_1\rangle$, and the unitary matrix H, which contained information about the diagonal basis of ρ. It is possible to formulate a fully quantum notion of type classes if we allow ourselves two variables to describe the type of quantum string, and expand our notion of type classes. Recall that classical type classes were closed under permutations. The Hilbert space $\mathcal{H}$ of possible states of a string of N qubits is 2^N-dimensional, and we can represent permutations as operators acting on this space. It turns out we can divide $\mathcal{H}$ into subspaces closed under the action of permutations as well as another set of operators called collective unitary rotations, which describe the same change of basis $\{|0\rangle, |1\rangle\} \to \{|0'\rangle, |1'\rangle\}$ applied to each qubit. When we tried to generalize classical types to the quantum case, we ran into problems because our naive approach was basis dependent. Therefore we look for subspaces which are not only closed under permutations but also under collective changes of basis.

For two qubits (which have a four dimensional Hilbert space), these subspaces are the one dimensional space spanned by the singlet

$$\left\{ \frac{|01\rangle - |10\rangle}{\sqrt{2}} \right\} \tag{18}$$

and the three-dimensional space spanned by the triplet

$$\left\{ |00\rangle, \frac{|01\rangle + |10\rangle}{\sqrt{2}}, |11\rangle \right\}. \tag{19}$$

Each subspace is closed under exchange of the qubit labels, with the singlet getting mapped to -1 times itself, and all states spanned by the triplet getting mapped to themselves. This is a familiar example from the study of spin angular momentum: the singlet is antisymmetric and the triplet is symmetric under exchange. One can easily check that each of these subspaces is closed under collective change of basis.

For example, if we try to express the singlet in the $\{|+\rangle, |-\rangle\}$ basis, we

obtain

$$\begin{aligned}\frac{|01\rangle - |10\rangle}{\sqrt{2}} \rightarrow & \frac{|+-\rangle - |-+\rangle}{\sqrt{2}} \\ &= \frac{(|0\rangle + |1\rangle)(|0\rangle - |1\rangle) - (|0\rangle - |1\rangle)(|0\rangle + |1\rangle)}{\sqrt{2}} \\ &= \frac{-|01\rangle + |10\rangle}{\sqrt{2}}\end{aligned} \tag{20}$$

which is still in the one-dimensional subspace spanned by the singlet. The same holds true for any local unitary transformation on either subspace. We can completely describe the type class of any two qubit state with the singlet and the triplet; the span of their union is the entire Hilbert space of two qubits.

Whereas classical type classes are sets of strings closed under permutations, quantum type classes are subspaces closed under permutations and system-wide changes of basis. Type (which in the classical case is just a number) is more complicated in the quantum case and involves both a number and information about the diagonal basis, as discussed at the beginning of this section. The quantum "size" of type classes is also more complicated than in the classical case. A full definition is beyond the scope of this paper, but see Blume-Kohout et. al. [8] for a detailed discussion. For the purposes of this paper, working in the diagonal basis of ρ will be sufficient to apply classical ideas of type to quantum computation.

1.2.3 Entanglement

We have discussed quantum message sources that output states $|\psi_i\rangle$ of individual qubits. We have dealt with multiple qubits in considering long strings of output from i.i.d. sources, but each qubit by the definition of its source was independent from all the rest. However, we can also have an i.i.d. source of quantum states in which each message consists of multiple qubits. While each multi-qubit state may be independent of the other, within each multi-qubit state, individual qubits may not be independent of one another. In particular, the qubits may be entangled.

Assume two parties, Alice and Bob, each have one qubit. If those qubits are in pure states $|A\rangle$ and $|B\rangle$, then the joint state of the system is just

$$|\Psi\rangle_{AB} = |A\rangle \otimes |B\rangle, \tag{21}$$

also known as a product state, and measuring one qubit will not affect the other. In this case there is no entanglement between Alice's qubit and Bob's.

The system may also be in some state

$$|\Psi\rangle_{AB} = \sqrt{p_1}\,|A_1\rangle \otimes |B_1\rangle + \sqrt{p_2}\,|A_2\rangle \otimes |B_2\rangle. \tag{22}$$

This is an entangled state of two qubits. If Alice measures her qubit and finds it is in state $|A_i\rangle$, then Bob's qubit immediately collapses to the state $|B_i\rangle$.

Even if Alice and her qubit are separated by a great distance from Bob and his qubit, any measurement of Alice's qubit will affect Bob's. Qubits are said to be entangled whenever this is the case.

The mathematical tool we usually employ in discussions of entanglement is the reduced density matrix. The reduced density matrix is a means of analyzing a particular subsystem, say A or B, of a larger quantum system AB. We obtain it by taking the partial trace, which is defined (following [7]) by

$$\begin{aligned} \mathrm{Tr}_A\Big(|a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2| \Big) &= |b_1\rangle\langle b_2| \langle a_1|a_2\rangle , \\ \mathrm{Tr}_B\Big(|a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2| \Big) &= |a_1\rangle\langle a_2| \langle b_1|b_2\rangle . \end{aligned} \tag{23}$$

Let's take as an example the two-qubit entangled state

$$\left|\Phi^+\right\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}, \tag{24}$$

also known as an "EPR pair" [1]. Its density operator is

$$\begin{aligned} \rho &= \left|\Phi^+\right\rangle\left\langle\Phi^+\right| \\ &= \left(\frac{|00\rangle + |11\rangle}{\sqrt{2}}\right)\left(\frac{\langle 00| + \langle 11|}{\sqrt{2}}\right) \\ &= \frac{|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|}{2} \\ &= \begin{pmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{pmatrix}. \end{aligned} \tag{25}$$

Note that $\mathrm{Tr}(\rho^2) = 1$, indicating ρ is a pure state. We can find the reduced density matrix of the first qubit by tracing out the second qubit,

$$\begin{aligned} \rho_1 = {} & \mathrm{Tr}_2(\rho) \\ &= \frac{\langle 0|0\rangle\,|0\rangle\langle 0| + \langle 0|1\rangle\,|0\rangle\langle 1| + \langle 1|0\rangle\,|1\rangle\langle 0| + \langle 1|1\rangle\,|1\rangle\langle 1|}{2} \\ &= \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \\ &= \frac{I}{2} \end{aligned} \tag{26}$$

where I represents the identity matrix. For this particular state, the reduced density matrix of the second qubit is also $\frac{I}{2}$. $\mathrm{Tr}(\frac{I}{2}^2) = \mathrm{Tr}(\frac{I}{4}) = \frac{1}{2}$, so ρ_1 and ρ_2 are mixed. They represent quantum message sources emitting $|0\rangle$ and $|1\rangle$ with equal probability. Although we have just argued that measurements of both qubits in an entangled state are not independent, in this case if we focus

on one individual qubit and ignore the other, we see its value has a probability distribution that greatly resembles that of an i.i.d. source. In fact, one way of creating a quantum i.i.d. message source is to prepare many identical copies of an entangled state and only consider one of its subsystems ([7] p. 542).

Another way of describing the properties of composite quantum systems, related to the reduced density matrix, is the Schmidt decomposition ([7] p. 109). If we have a system AB composed of subsystems A and B in state $|\psi\rangle$, then we can always find orthonormal states $|i_A\rangle$ for system A and $|i_B\rangle$ for system B such that

$$|\psi\rangle = \sum_i \lambda_i |i_A\rangle \otimes |i_B\rangle . \tag{27}$$

In the Schmidt decomposition, because $|i_A\rangle$ and $|i_B\rangle$ are each orthonormal, the reduced density matrices for A and B are

$$\rho_A = \sum_i \lambda_i^2 |i_A\rangle \langle i_A| \tag{28}$$

and

$$\rho_B = \sum_i \lambda_i^2 |i_B\rangle \langle i_B|. \tag{29}$$

These density matrices are both diagonal. Thus the $\{|i_A\rangle\}$ basis and the $\{|i_B\rangle\}$ basis we find in the Schmidt decomposition are the diagonal bases for systems A and B respectively, and furthermore they share the same eigenvalues λ_i^2. $\{|i_A\rangle\}$ and $\{|i_B\rangle\}$ are known as Schmidt bases.

For two qubits, although $|i_A\rangle$ and $|i_B\rangle$ are not in general equivalent, since both are sets of orthogonal vectors in 2-dimensional complex Hilbert spaces, we can rotate them separately until they are the same. In other words, we can apply the operator $U_A \otimes U_B$ to $|\psi\rangle$ in eq. 27, mapping both $|i_A\rangle$ and $|i_B\rangle$ to equivalent, orthogonal states $|0'\rangle$ and $|1'\rangle$. Since the trace of a density matrix is invariant under unitary operations, our change of basis does not change the entanglement of $|\psi\rangle$. Therefore, with the proper unitary transformations, we can express any two qubit state $|\psi\rangle$, without altering its entanglement, as

$$|\psi\rangle = \lambda_0 |0'0'\rangle + \lambda_1 |1'1'\rangle . \tag{30}$$

For two qubits, we usually refer to $\{|0'\rangle , |1'\rangle\}$ as the Schmidt basis, even though there are technically two Schmidt bases $\{|i_A\rangle\}$ and $\{|i_B\rangle\}$ that we rotate with U_A and U_B to obtain eq. 30. Although we do not necessarily know which unitary transformations U_A and U_B we should apply, it is possible to estimate them with minimal disturbance to $|\psi\rangle$ [8]. Therefore in many algorithms involving two qubit states, we may assume without loss of generality that we operate in the Schmidt basis.

1.2.4 Von Neumann Entropy

We can quantitatively describe both entanglement and the properties of quantum message sources with the von Neumann entropy. The von Neumann entropy

is

$$S(\rho) = -\mathrm{Tr}(\rho \log \rho). \tag{31}$$

In terms of the eigenvalues of ρ, λ_i, we can rewrite this as

$$S(\rho) = -\sum_i \lambda_i \log \lambda_i, \tag{32}$$

which is equal to the classical Shannon entropy of an i.i.d. source emitting different values i with probability λ_i. We will find that we can adapt many algorithms from classical information theory (especially those involving type classes) to work with qubits simply by changing our computational basis to the eigenbasis of ρ and implementing all operations with quantum logic gates.

We can measure entanglement between qubits in system A by taking the von Neumann entropy of the density matrices describing the subsystems of A. Let's return to our EPR pair example, $|\Phi^+\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$, $\rho_1 = \rho_2 = \frac{I}{2}$. If we trace out the second qubit we find the reduced density matrix of the first, $\frac{I}{2}$. The von Neumann entropy of either subsystem is

$$S\left(\frac{I}{2}\right) = -\frac{1}{2}\log\left(\frac{1}{2}\right) - \frac{1}{2}\log\left(\frac{1}{2}\right) = \log(2) = 1 \text{ bit}. \tag{33}$$

Note that the entropy, which for a two-outcome source varies from 0 to 1, is maximal in this case. Maximal uncertainty about one quantum subsystem in absence of information about the other corresponds to maximal entanglement. The first and second qubits of $|\Phi^+\rangle$ are maximally entangled, and so we call $|\Phi^+\rangle$ a maximally entangled state. Note however that the von Neumann entropy is only a meaningful entanglement measure for the state $|\Phi^+\rangle$ because the entanglement of its first qubit with its second is equal to the entanglement of its second with its first. In general, the entanglement of subsystem A with subsystem B as measured by the von Neumann entropy of its reduced density matrix will always be equal to the entanglement of subsystem B with subsystem A, which we can prove using the Schmidt decomposition. For states of two qubits, we define the entanglement E of the state to be equal to this entropy.

The two senses in which we have used the von Neumann entropy–as a measure of the information we gain from a quantum message source and as a measure of the entanglement between two systems–parallel the two senses in which we interpreted the classical Shannon entropy. Our consideration of the Shannon entropy as quantifying the information we gain after measuring a random variable X is analogous to the von Neumann entropy as quantifying the information in data from a quantum message source. Problems we will want to consider from this perspective include the compression of quantum and classical information. Our consideration of the Shannon entropy as quantifying X's "randomness" or our uncertainty about X before we measure it parallels our discussion of the von Neumann entropy and entanglement. The most entangled subsystems are those we know least about in absence of information about the others. When we attempt to solve the quantum problem of entanglement concentration, the

classical tools we will seek to adapt to our uses will be randomness extraction algorithms.

In this section we have introduced a number of theoretical ideas to help us describe message sources of classical and quantum information and various properties of their outputs. In the following sections, we will make use of those properties to formulate algorithms for data compression (Sections 2 and 3) and entanglement concentration (Section 4). Within each of these sections, we will discuss different systems, which will necessarily require we come up with different algorithms. In Section 2.1 we will consider how to compress strings of bits from an i.i.d. source and in Section 2.2 we will consider strings of qubits from an i.i.d. source. In Section 3 we will consider how to apply the ideas of compression to physical systems, specifically data obtained from configurations of grand canonical ensembles of bosons and fermions, which are not i.i.d. sources. In Section 4.1 we will consider entanglement concentration for sources emitting states of two qubits, and in Section 4.2 we will consider sources that emit three-qubit states and some of the challenges we run up against in trying to concentrate entanglement from such sources.

2 Data Compression

2.1 Classical Compression of i.i.d. Bit Strings

We now describe an algorithm, outlined in the introduction and in [10], that compresses sequences of classical bits from i.i.d. sources by ordering them from most to least likely and only counting those which are typical. The algorithm operates as follows. Assume that our source emits outcome "0" more often than "1." If the opposite is true we can flip all bits upon acquisition. Bit strings of length N are sorted into bins based on their type T. The bit string with $T = 0$ will be assigned to the lowest bin, which has size $\binom{N}{0} = 1$ and has starting address 0. All bit strings of type 1 will be assigned to the second lowest bin, which has size $\binom{N}{1} = N$ and starting address $\binom{N}{0} = 1$. Bit strings with two 1s will be assigned to the third lowest bin, which has size $\binom{N}{2} = \frac{N(N+1)}{2}$ and starting address $\binom{N}{0} + \binom{N}{1}$... and so on. Within each of those bins, we index bit strings by interpreting them as binary representations of integers and ordering those integers by value. The lowest valued strings are given the lowest intrabin indices. Ultimately, each string receives a total mapping number which equals its intrabin index plus the address where its bin begins.

For example, the set of 4-bit strings gets mapped as shown in Figure 2. In general, for length N strings, we need $N + 1$ bins with sizes given by $\binom{N}{T}$ to count all possible combinations. Regardless of how we index within bins, if we start each bin where the previous one left off the value of the first string in each bin of type T will be the sum of the sizes of all bins with type less than T, or

$$\sum_{i=0}^{T-1} \binom{N}{i}. \tag{34}$$

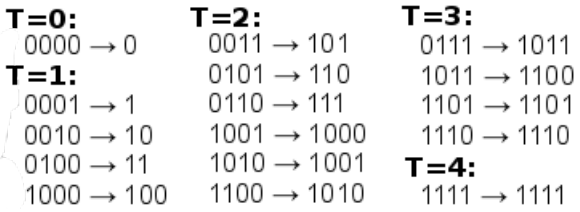

Figure 2: 4-bit mapping. Bitstrings of length 4 are mapped to one of five differently sized bins depending on their type (0, 1, 2, 3, or 4). Within each bin they are ordered by total binary value. We then assign each string an index by counting them, and for compression, replace each string with its index.

Therefore, all bit strings of length N and type T get mapped to numbers between

$$\sum_{i=0}^{T-1} \binom{n}{i}$$

and

$$\sum_{i=0}^{T} \binom{n}{i} - 1,$$

inclusive.

Each bin contains a set of bit strings. To index them, we decide to interpret them as digits of integers represented in base 2, and then we order the integers. We construct a function $I[x, N, T]$ that assigns every string in the bin a unique index number from 0 to $\binom{N}{T}$-1, with the strings representing the lowest binary values being assigned the lowest indices.

Sequences with low binary values have most of their zeroes in their most significant digits. We can calculate I in stages by counting the number of *leading zeroes* in each sequence, and then dividing it into subsequences. I is simply and most easily expressed recursively:

$$I[x, N, T] = \binom{N-p-1}{T} + I[x', N-p-1, T-1], \tag{35}$$

where p is the number of leading zeroes in the bit string x, and x' is what remains of x after the p leading zeroes and the first leading one are thrown out. Cleve and DiVincenzo show that I assigns every sequence a unique value ranging from 0 to $\binom{N}{T}$-1 based on the value of the integer it represents [2].

Two quick examples just for illustration:

$$\begin{aligned} I[1011,4,3] &= \binom{4-0-1}{3} + I[011,3,2] \\ &= \binom{3}{3} + \binom{3-1-1}{2} + I[1,1,1] \\ &= \binom{3}{3} + \binom{1}{2} + \binom{0}{1} = 1. \end{aligned} \tag{36}$$

Referring back to Fig. 2, we see that, within the $T = 3$ bin, string 1011 is indexed second, which corresponds to $I = 1$ since we count from $I = 0$. We obtain the mapping number of 1011, which is $12 = 1100$, by adding this index, 1, to the start of the bin, $\binom{4}{0} + \binom{4}{1} + \binom{4}{2} = 11$.

Second example:

$$\begin{aligned} I[0110,4,2] &= \binom{4-1-1}{2} + I[10,2,1] \\ &= \binom{2}{2} + \binom{2-0-1}{1} + I[0,1,0] \\ &= \binom{2}{2} + \binom{1}{1} + \binom{-1}{0} = 2 \end{aligned} \tag{37}$$

Within the $T = 2$ bin, string 0110 is indexed third, which corresponds to $I = 2$ as shown here. We obtain the mapping number of 0110, which is $7 = 111$, by adding $I = 2$ to the start of the bin, $\binom{4}{0} + \binom{4}{1} = 5$. Note that in this second example, where our initial string had $T = 2$ rather than $T = 3$, we needed only 3 bits to represent its mapping number.

This algorithm maps strings of bits to numbers based on their type and binary value. There are 2^N possible strings of length N, so in general we will need N bits to count all of them, but if the i.i.d. source X from which we draw our bits has entropy $H(X)$ then there are only $2^{NH(X)}$ typical sequences we need to cover. Assuming our source emits 0 preferentially, all of the likely strings will have types less than than those of the unlikely strings. Our algorithm by design maps lower types to lower numbers, which if expressed in N bits will have $N - NH(X)$ leading zeroes. To compress our output, we can discard all but the $NH(X)$ least significant bits. The discarded bits should all be zero if we have drawn a likely string. To decompress our data, we can append our string to $N-NH(X)$ zeroes and run the algorithm in reverse. We do not need to store those zeroes along with our compressed result to reconstruct it with accuracy. Thus, assuming we have knowledge of $NH(X)$ and whether our source emits 0 or 1 preferentially, we can store length N sequences in $NH(X)$ bits.

However, we do not always have such knowledge of our source. In situations where information about the entropy is unavailable, we must turn to universal compression algorithms, which are defined as algorithms that can compress data from i.i.d. sources from N bits to $NH(X)$ bits without prior knowledge of$H(X)$. One such algorithm is known as Lempel-Ziv compression [13], which we discuss in Appendix A.

2.2 Generalization to Strings of Qubits

Cleve and DiVincenzo [2] present a quantum implementation of this algorithm, first suggested in the quantum case by Schumacher [9]. Their algorithm performs the compression outlined above on computational basis states. While inputs may in general be superpositions of computational basis states, by the linearity of the algorithm describing the its effect on a complete set of basis states is sufficient to describe its effect on all inputs. The Schumacher compression protocol takes basis states of N qubits, of which T are $|1\rangle$ and $N-T$ are $|0\rangle$, and transforms them to other basis states representing their mapping numbers. The algorithm maps all basis states with T $|1\rangle$s to states between

$$\left|\sum_{i=0}^{T-1}\binom{n}{i}\right\rangle$$

and

$$\left|\sum_{i=0}^{T}\binom{n}{i}-1\right\rangle,$$

inclusive, and within those bins we index states by interpreting them as binary numbers and ordering them by value.

The quantum algorithm is a direct translation of the classical one into quantum logic gates. We are able to do this because we work in a basis in which we assume all our qubits are $|0\rangle$ or $|1\rangle$. The algorithm is made universal for strings of qubits from any i.i.d. source ρ by assuming our ability to change to this basis, which we have argued is the diagonal basis of ρ. Our input is always a product of basis states in the diagonal basis of ρ. However, if we did give our algorithm an input in a superposition of computational basis states, our output would just be a superposition of corresponding output states by the linearity of all the operators we will employ.

Cleve and DiVincenzo present the following code, which I will briefly explain:

Schumacher Compression Protocol
`quantum registers`:
X : n-`bit signed arithmetic register` //X holds string and result
T : $\log(2n)$-`bit arithmetic register initialized and finalized by 0`
//T holds type

1. `for` $p = n-1$ `down to 0 do` $\{$
2. `if` $X_{n-p-1} = 1$ `then` $T \leftarrow T+1$ //set T to type of X
3. `for` $i = n-p$ `down to 0 do` $\{$
4. `if` $T = i$ `and` X_{n-p-1} `then` $X \leftarrow X + \binom{n-p-1}{i}$
//add index number to X

```
5.          if T = i and TRUNC_{n-p-1}(X) ≥ C(n-p-1, i)
            then X_{n-p-1} ← X_{n-p-1} ⊕ 1        //delete original content of X
6.        }
7.    }                                           //by this point, X = I(X, N, T)
8.    for m = n down to 0 do {
9.        if T ≤ m then X ← X − C(n, m)
10.   }
11.   for m = n down to 0 do {
12.       if X ≥ 0 then T ← T − 1                 //reset T to 0
13.       X ← X + C(n, m).                        //add bin address to X
14.   }
```

There are three main loops in this code, all of which loop over the entirety of the string. The first loop (line 1) calculates the type by adding 1 to the ancilla register T (line 2) for every $|1\rangle$ in the input. We start at the least significant digit of X and loop up.

It then goes into a subloop (line 3) that transforms X into a state representing its index number I, bit by bit. Recall that

$$I[x, n, t] = \binom{n-p-1}{t} + I[x', n-p-1, t-1],$$

where p is the number of leading zeroes in the bit string x, and x' is what remains of x after the p leading zeroes and the first leading one are thrown out, and n and t represent the length and type of the string. I is a sum of increasingly small terms that represent the index numbers of smaller and smaller portions of the string. Line 4 adds terms of this sum, with the smallest first, to X. Line 5 goes through and makes sure the bits of X we no longer need to calculate I are set to 0, such that X has been transformed to I by line 7.

The loops at lines 8 and 11 together add the bin address to X such that it becomes the mapping number. Thus the algorithm performs the transformation in-place. In the process it resets our ancilla register T to 0 so that our output is not entangled with it. It adds this address by adding all possible $\binom{n}{m}$ terms to get the maximum address, and conditionally subtracting the highest terms of of that address to get the actual address. It does the subtraction first because it simplifies the resetting of T. If T is close to n then we do almost no subtraction at line 9, so X becomes positive very quickly and we subtract from T many times in line 12 to reset it to 0. If T is close to 0, then we do quite a lot of subtraction at line 9, and very little subtraction from T in line 12.

The length of X times the entropy $S(\rho)$ of the i.i.d. source ρ from which we draw X tells us how many bits of our result, starting from the least significant, are likely to be nonzero, as $n \to \infty$. If we know $S(\rho)$, then from here we can simply discard the $n - nS(\rho)$ most significant bits of our transformed X, which should all be 0. We can measure them to make sure. Then we store our compressed string until we wish to decompress it, at which point we append our string to $n - nS(\rho)$ copies of $|0\rangle$ and run the decompression protocol. Because the compression protocol presented in this section is by design entirely reversible, the decompression protocol is just its time-reverse. See [2] for more details.

3 Application of Compression to Physical Systems

Thus far our discussion has focused on things to do with data from i.i.d. message sources. However, most quantum systems of interest are neither independent nor identically distributed. Stanislaw Ulam once remarked that non-linear physics "is like non-elephant biology." We can say much the same thing for non-i.i.d. quantum information theory.

Johnson and Suhov [3, 4, 5, 6] have begun to expand the field of quantum information theory to non-i.i.d. sources with applications to physical systems. In particular, they show [3] that for systems of non-interacting fermions and bosons, one can define and compute the entropy and compress information from those systems with a compression ratio given by that entropy. In the subsections below we will illustrate some of their conclusions for two familiar thermodynamic systems: bosons and fermions in a grand canonical ensemble.

3.1 Independent but not Identically Distributed Case– Grand Canonical Ensemble

Consider a ensemble of noninteracting quantum particles for which the full system Hamiltonian, H, is a sum of the one-particle Hamiltonians H_1 over the total number of particles. The eigenstates of H_1, $|n\rangle$, have eigenvalues γ_n, where $0 \leq n < \infty$. To ground our system in a physical example we will assume the particles are in an infinite 1-D square well of length L, so that

$$\begin{aligned} H_1 &= -\frac{h^2}{2m}\nabla^2 + V(x) \\ V(x) &= 0 \quad \text{if } 0 \leq x \leq L, \\ &\infty \text{ otherwise} \end{aligned} \tag{38}$$

and

$$\gamma_n = \frac{n^2 h^2}{8mL^2}. \tag{39}$$

For this system, H_1 and geometry of the system completely describe the space of available system-wide configurations. The total number of particles is variable and the cost of adding a particle is given by the chemical potential μ. We assume the system is in thermal equilibrium with its surroundings at temperature T, so particles enter the system with thermal energy proportional to kT, or $1/\beta$. We construct a multi-particle state by independently assigning each particle to one of the one particle infinite square well eigenstates $|n\rangle$ (with fermionic or bosonic symmetry restrictions when they apply). We assume Gibbs statistics: that is, when we are assigning a particle, the probability p that it will fill state $|n\rangle$ is proportional to $e^{-\beta(\gamma_n+\mu)}$.

We are interested in states or configurations of the entire system of particles. These eigenstates are denoted $|\phi_{\mathbf{k}}\rangle$, and as the subscript implies they are associated with a particular configuration $\mathbf{k}$. $\mathbf{k} = \{k_n, 0 \leq n < \infty\}$ is a sequence of particle occupation numbers k_n across n, which can be arbitrarily long. For bosons, k_n can be any integer, so $\mathbf{k}$ is a string of integers. For fermions, k_n must be 0 or 1, making $\mathbf{k}$ a string of bits. Figure 3 provides some example configurations of bosons and fermions to illustrate these definitions. In a thermal state

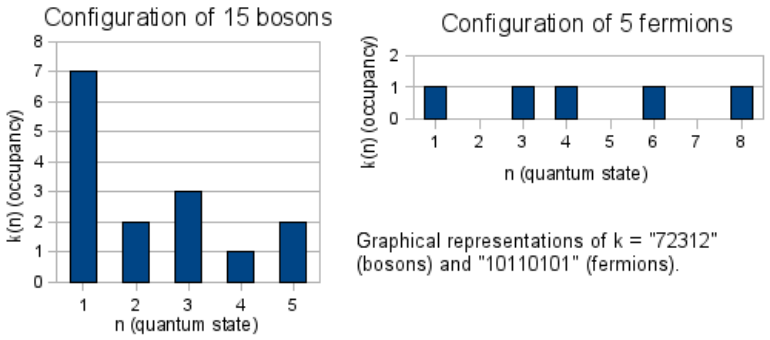

Figure 3: Examples of data represented by ensembles of bosons and fermions. Strings, represented by the bar graphs, are constructed by measuring the configuration of the ensemble.

characterized by a chemical potential μ and inverse temperature β, each multiparticle configuration labelled by $\mathbf{k}$, $|\phi_{\mathbf{k}}\rangle$, has a probability $P(\mathbf{k})$ associated with it, equal to

$$P(\mathbf{k}) = \exp\left(-\beta \sum_{\mathbf{n}\in N} k_n(\mu + \gamma_{\mathbf{n}})\right) = \prod_{\mathbf{n}\in N} \exp\left(-\beta(\gamma_{\mathbf{n}} + \mu)k_n\right). \qquad (1)$$

This probability is just obtained by calculating the total energy E of the system-wide eigenstate or configuration (summed over one-particle eigenstates, this just the chemical potential μ times the number of particles k_n in a one-particle eigenstate, plus that same number of particles times $\gamma_{\mathbf{n}}$, the energy per

particle), and then assuming that the probability of the system being in that configuration is proportional to $e^{-\beta E}$.

Note that the probability of our source outputting $\mathbf{k}$ is just the product of the probabilities that each individual particle fills each of its eigenstates on any particular assignment. Mathematically this indicates that our source is independent.

However, the probability that a particular eigenstate is filled when we decide to add a particle to the system ($e^{-\beta(\gamma_n+\mu)}$) is not the same as the probability of a particular state *being* filled if system is allowed to evolve freely at temperature T and the number of particles allowed to vary. The output of our message source is not whether we succeed at filling $|n\rangle$ on any one attempt but rather whether k_n *is* filled at thermal equilibrium. To determine whether or not the system is identically distributed we must determine this probability. The distribution is different for fermions and bosons so we will consider them separately.

3.2 Fermions

We know by eq. 1 that

$$P(\mathbf{k}) = \prod_{\mathbf{n}\in N} \exp\left(-\beta(\gamma_{\mathbf{n}}+\mu)k_n\right)$$

for a specific system configuration $\mathbf{k}$. We are looking for the probability that a specific one-particle eigenstate $|n'\rangle$ is filled; or rather, the probability $p_{n',1}$ that $k_{n'} = 1$ regardless of the configuration as a whole. We can get this by simply summing the probabilities of all system configurations $\mathbf{k}$ that have $k_{n'} = 1$:

$$p_{n',1} = \frac{\sum\limits_{\mathbf{k},\mathrm{k_{n'}}=1} P(\mathbf{k})}{\sum\limits_{\mathbf{k}} P(\mathbf{k})}, \tag{40}$$

where each $\sum P(\mathbf{k})$ by the independence of our source as given in eq. 1 is just a sum of products. Thus we can factor $\exp(-\beta(\gamma_{n'}+\mu)\times 1)$ out of the numerator to re-express this as

$$\frac{e^{-\beta(\gamma_{n'}+\mu)} \sum\limits_{\mathbf{k}} \left(\prod\limits_{n\neq n'} e^{-\beta(\gamma_n+\mu)k_n} \right)}{\sum\limits_{\mathbf{k}} \left(\prod\limits_{n} e^{-\beta(\gamma_n+\mu)k_n} \right)}, \tag{41}$$

and factoring out a similar expression from the denominator, we are left us with

$$\frac{e^{-\beta(\gamma_{n'}+\mu)} \sum\limits_{\mathbf{k}} \left(\prod\limits_{n\neq n'} e^{-\beta(\gamma_n+\mu)k_n} \right)}{\sum\limits_{k_{n'}=0}^{k_{n'}=1} e^{-\beta(\gamma_{n'}+\mu)k'_n} \sum\limits_{\mathbf{k}} \left(\prod\limits_{n\neq n'} e^{-\beta(\gamma_n+\mu)k_n} \right)}, \tag{42}$$

which cancels out to

$$p_{n',1} = \frac{e^{-\beta(\gamma_{n'}+\mu)}}{1+e^{-\beta(\gamma_{n'}+\mu)}} = \frac{1}{1+e^{\beta(\gamma_{n'}+\mu)}}, \tag{43}$$

also known as the Fermi-Dirac distribution. It is a well known result that the probability of eigenstates being filled in a fermionic system is given by the Fermi-Dirac distribution. Since this probability varies with n, our source is not identically distributed.

However, as we have already argued, our source ρ *is* independent, and we can use this property to find its entropy. To find $S(\rho)$ we need to quantify the information we gain from measuring a configuration $\mathbf{k}$. Measuring $\mathbf{k}$ consists of infinitely many independent measurement events k_n. The entropy of a sequence of independent events is always the sum of the entropies of each individual event. Therefore

$$S(\rho) = \sum_{n}^{\infty} S(k_n). \tag{44}$$

For any $|n\rangle$, the probability that we find it filled ($k_n = 1$) is $\frac{e^{-\beta(\gamma_{n'}+\mu)}}{1+e^{-\beta(\gamma_{n'}+\mu)}}$. The probability that we find it empty is $1 - \frac{e^{-\beta(\gamma_{n'}+\mu)}}{1+e^{-\beta(\gamma_{n'}+\mu)}} = \frac{1}{1+e^{-\beta(\gamma_{n'}+\mu)}}$. Defining the Gibbs factor

$$q(n) \equiv e^{-\beta(\gamma_{n'}+\mu)}, \tag{45}$$

we can write down the entropy S of this measurement event

$$S(k_n) = -\frac{q(n)}{1+q(n)} \log\left(\frac{q(n)}{1+q(n)}\right) - \frac{1}{1+q(n)} \log\left(\frac{1}{1+q(n)}\right). \tag{46}$$

By eq. 44, the entropy of the entire fermionic ensemble ρ is

$$S(\rho) = -\sum_{n=0}^{\infty} \frac{q(n)}{1+q(n)} \log\left(\frac{q(n)}{1+q(n)}\right) + \frac{1}{1+q(n)} \log\left(\frac{1}{1+q(n)}\right). \tag{47}$$

Although we can obtain a close approximation of this entropy by simply terminating the sum at large n, it is useful to express this sum as an integral. Additionally, it is useful to write down a function for the entropy per unit length of the system, rather than the entropy itself, to reduce the total number of parameters, which Johnson and Suhov [3] denote h_- for fermions. Since we have chosen to place our ensemble in an infinite square well, $\gamma_n = \frac{n^2h^2}{8mL^2}$. We can define

$$y \equiv \frac{n}{L} \tag{48}$$

and re-express γ_n as $\gamma_y = \frac{h^2}{8m}y^2$, and $q(n)$ as $q(y)$. Then the entropy per unit length is

$$h_- = -\sum_{n=0}^{\infty} \frac{q(y)}{1+q(y)} \log\left(\frac{q(y)}{1+q(y)}\right) + \frac{1}{1+q(y)} \log\left(\frac{1}{1+q(y)}\right)\frac{1}{L}. \tag{49}$$

Since $\Delta n = 1$, we can rewrite $\frac{1}{L} = \frac{\Delta n}{L} = \Delta y$, and express our sum as

$$h_- = -\sum_{y=0}^{\infty} \frac{q(y)}{1+q(y)} \log\left(\frac{q(y)}{1+q(y)}\right) + \frac{1}{1+q(y)} \log\left(\frac{1}{1+q(y)}\right)\Delta y. \qquad (50)$$

If we take the limit as $L \to \infty$, then y becomes a continuous variable, and eq. 50 is a Riemann sum converging to

$$h_- = -\int_0^{\infty} \frac{q}{1+q} \log\left(\frac{q}{1+q}\right) + \frac{1}{1+q} \log\left(\frac{1}{1+q}\right) dy, \qquad (51)$$

which Johnson and Suhov [3] assume is finite.

The essence of Shannon or Schumacher's theorem, expressed in terms of this example, is that if we consider a random configuration $\mathbf{k}$ in a well of length L, then as $L \to \infty$ it will almost surely be in the likely set. Eq. 51 describes the minimum number of bits needed to count all likely configurations of an ensemble of fermions in an infinite square well, divided by the length of that square well. Because eq. 51 describes the greatest lower bound on the computational resources needed to represent those configurations, it also quantifies our uncertainty about a random configuration and the information we gain by measuring it.

3.2.1 Bosons

As for fermions, the same formula

$$P(\mathbf{k}) = \prod_{\mathbf{n} \in N} \exp\left(-\beta(\gamma_{\mathbf{n}} + \mu)k_n\right)$$

holds for bosons, but this time k_n can be any integer. We know our source is independent, but what is the probability $p_{n',x}$ that there are $k_{n'} = x$ bosons in state $|n'\rangle$ regardless of the configuration of the rest of the system? Adopting the same strategy for bosons as for fermions, we can get this by simply summing the probabilities of all configurations $\mathbf{k}$ that have $k_{n'} = x$:

$$p_{n',x} = \frac{\sum_{\mathbf{k},\mathrm{k_{n'}=x}} P(\mathbf{k})}{\sum_{\mathbf{k}} P(\mathbf{k})}, \qquad (52)$$

which we can express as

$$\frac{e^{-\beta(\gamma_{n'}+\mu)x}\sum_{\mathbf{k}}\left(\prod_{n\neq n'}e^{-\beta(\gamma_n+\mu)k_n}\right)}{\sum_{\mathbf{k}}\left(\prod_{n}e^{-\beta(\gamma_n+\mu)k_n}\right)}$$

$$=\frac{e^{-\beta(\gamma_{n'}+\mu)x}\sum_{\mathbf{k}}\left(\prod_{n\neq n'}e^{-\beta(\gamma_n+\mu)k_n}\right)}{\sum_{k_{n'}=\infty}^{k_{n'}=0}e^{-\beta(\gamma_{n'}+\mu)k'_n}\sum_{\mathbf{k}}\left(\prod_{n\neq n'}e^{-\beta(\gamma_n+\mu)k_n}\right)}, \quad (53)$$

which cancels out to

$$p_{n',x}=\frac{e^{-\beta(\gamma_{n'}+\mu)x}}{\sum_{k_{n'}=0}^{\infty}e^{-\beta(\gamma_{n'}+\mu)k_{n'}}}. \quad (54)$$

Again defining $q(n)\equiv\exp(-\beta(\gamma_{n'}+\mu))$, we can write this as

$$p_{n',x}=\frac{q(n)^x}{\sum_{k_{n'}=0}^{\infty}q(n)^{k_{n'}}}, \quad (55)$$

which is a series expansion for

$$(1-q(n))q(n)^x, \quad (56)$$

also known as the geometric distribution. Since the probability of occupancy is again dependent on n, we can conclude that a grand canonical ensemble of bosons is also an independent but not identically distributed message source with a different distribution function than that of fermions.

Calculating the entropy of a random bosonic configuration is slightly more complicated, because instead of two possible outcomes ($k_n=0$ and $k_n=1$) for every measurement event, we have infinitely many outcomes since x can be any non-negative integer. The entropy at any n is

$$S(k_n)=-\sum_{x=0}^{\infty}(1-q)q^x\log\left((1-q)q^x\right). \quad (57)$$

Expanding the first few terms,

$$\begin{aligned}&(1-q)\log(1-q)\\&+(1-q)q\log((1-q)q)\\&+(1-q)q^2\log((1-q)q^2)\\&+(1-q)q^3\log((1-q)q^3)\cdots\end{aligned} \quad (58)$$

we see that we can re-express this as

$$-S(k_n) = (1-q)(1+q+q^2+q^3\cdots)\log(1-q) \\ +q(1-q)(1+2q+3q^2+4q^3\cdots)\log(q). \quad (59)$$

Noting that $0 < q < 1$ and defining $1 < t \equiv \frac{1}{q}$, we can express the first infinite series $(1+q+q^2+\cdots)$ as

$$\sum_{i=0}^{\infty} \frac{1}{t^i}, \quad (60)$$

which converges to

$$1 + \frac{1}{t-1} = \frac{t}{t-1} = \frac{1}{1-q}. \quad (61)$$

The second infinite series $(1+2q+3q^2+\cdots)$ is just the derivative of the first with respect to q, so it converges to

$$\frac{d}{dq}\left(\frac{1}{1-q}\right) = \frac{1}{(1-q)^2}. \quad (62)$$

Therefore, the entropy of any measurement of k_n for a bosonic configuration is

$$S(k_n) = -\log(1-q) - \frac{q}{1-q}\log(q) = -\frac{q\log(q)+(1-q)\log(1-q)}{1-q}. \quad (63)$$

Even though our set of possible outcomes is infinitely large, this expression looks a bit like the Shannon entropy of a two-outcome source. However, the probabilities of these "outcomes" are actually the probabilities of filling or not filling a particular one-particle state n when we add a new particle to the system.

Because each k_n measurement is independent, the total entropy of the system, i.e. the information we obtain by measuring the entire configuration $\mathbf{k}$, is

$$S(\rho) = \sum_{n=0}^{\infty} S(k_n) = -\sum_{n=0}^{\infty} \frac{q(n)\log(q(n)) + (1-q(n))\log(1-q(n))}{1-q(n)}. \quad (64)$$

Following the same line of argument we used for fermions, we can define $y \equiv \frac{n}{L}$ and rewrite eq. 64 as a Riemann sum converging to

$$h_+ = -\int_0^{\infty} -\frac{q\log(q)+(1-q)\log(1-q)}{1-q}\,dy. \quad (65)$$

Eq. 65 describes the greatest lower bound on the computational resources needed to represent configurations $\mathbf{k}$ of a grand canonical ensemble of non-interacting bosons, as $L \to \infty$.

3.3 How to Compress Grand Canonical Ensemble Data

It is worth taking a moment to think more about the data we receive as output from this system, both for fermions and for bosons, and how we might compress it. We know we receive a string of bits in the fermionic case and a string of integers in the bosonic case. Since there are an infinite number of quantum states $|n\rangle$ in an infinite square well, our string is technically of infinite length. However, since γ_n increases like n^2, the probability of finding $k_n \neq 0$ at $|n\rangle$ is extremely low for very large n, so we may be able to truncate the string representing our configuration to a finite length.

The question of compression then is perhaps twofold: first, at what value of n can we terminate our string and be more or less assured of not losing any data? Secondly, once we do terminate our string, can we compress it any further? For example, if the chemical potential μ is very large compared to the gap between γ_n, then we are likely to have a particle-sparse system, where very few $|n\rangle$ will be occupied almost regardless of n. Even though the string "$0\cdots1000$" will still be several times less likely than "$0\cdots0001$," both will be much more likely than "$0\cdots0011$." If this effect is pronounced enough, then the major determinant of probability will be total number of particles, which in the fermionic case is just the classical type of the string. In this case we may want to simply apply Schumacher-style quantum compression, which depends on a useful correspondence between probability and type, to our data. However, if the unidentically distributing factor γ_n is significant compared to μ, the probability may not depend strongly on the number of particles, and we may opt for a different compression scheme based, say, on total binary value. Compression *is* theoretically possible, though: Johnson and Suhov suggest a quantum implementation of Lempel-Ziv compression (see Appendix A), and go on to show that Lempel-Ziv compresses data from this non-i.i.d. source with a compression ratio equal to the von Neumann entropy. This proof is beyond the scope of the present paper [3].

Practically, the question of how to actually realize such algorithms is also beyond the scope of this paper. However, we could imagine some situation where we measured the occupation numbers of an ensemble in a particular potential and stored those results on a qubit-based quantum computer, without any measurement. We might find it very useful to compress that data without disturbing its state. The questions we have considered in this section of the paper apply directly to such situations. At the very least we have made progress in describing the information-theoretic properties of non-i.i.d. sources.

4 Entanglement Concentration

We have now considered the information-theoretic properties of two message sources that are independent but not identically distributed. In the following sections we will consider message sources that are not independent. Our focus in these sections will not be on data compression but on entanglement

concentration. Data compression is the act of storing a particular amount of information spread across N physical systems in a smaller number of physical systems $NH(X)$. Entanglement concentration is the act of storing a particular amount of entanglement between subsystems A and B of ρ, spread across N copies of ρ, in a smaller number $NS(\rho^A) = NS(\rho^B)$ of maximally entangled systems. These ideas are closely related, but the classical algorithms we will want to generalize in this case will not be ones of data compression but of randomness extraction.

4.1 2 Qubit Entanglement

We frame the general problem of two-qubit entanglement concentration as follows. Alice and Bob have a source of N identical states $\sqrt{p_1}\,|A_1\rangle \otimes |B_1\rangle + \sqrt{p_2}\,|A_2\rangle \otimes |B_2\rangle$, each of entanglement $E < 1$ bits. They wish to convert these qubits via local operations to NE copies of the EPR state $\frac{|0\rangle|0\rangle+|1\rangle|1\rangle}{\sqrt{2}}$, each of entanglement 1 bit.

Without loss of generality we will assume Alice and Bob operate in the Schmidt basis $\{|0'\rangle, |1'\rangle\}$ where their states are of the form

$$\sqrt{p}\,|0'0'\rangle + \sqrt{1-p}\,|1'1'\rangle,$$

each of entanglement $H(p)$ bits. For such a state, the reduced density matrices of each qubit are both $p\,|0'\rangle\langle 0'| + (1-p)\,|1'\rangle\langle 1'|$. This means that Alice and Bob effectively have i.i.d. sources of perfectly correlated qubits. Each of them can then run randomness extraction algorithms on their qubits, which take streams of input from sources of the form $p\,|0\rangle\langle 0| + (1-p)\,|1\rangle\langle 1|$ and output qubits as if they were from a source $\frac{|0\rangle\langle 0|+|1\rangle\langle 1|}{2}$. Recall that this is the reduced density matrix for either qubit of an EPR pair. If Alice and Bob both run randomness extraction algorithms on their qubits, which are perfectly correlated in the Schmidt basis, then they can concentrate their states to EPR pairs.

4.1.1 Von Neumann's Randomness Extraction Protocol

Von Neumann's classical randomness extraction protocol [11] is a method a single actor, Chester, employs to generate uniformly random bits (prob(0) = 50% = prob(1)) from an i.i.d. source of bits with prob(0) = p and prob(1) = $1-p$. Von Neumann's quantum protocol is a method Alice and Bob employ to generate EPR pairs from a source of identical states $|\psi\rangle = \sqrt{p}\,|0_A 0_B\rangle + \sqrt{1-p}\,|1_A 1_B\rangle$.

Von Neumann's classical protocol operates as follows. Chester reads in two random bits from the i.i.d. source and checks their parity. If it is 1, then he knows he has drawn either "01" or "10." Chester outputs the first bit. Since the source is i.i.d., strings of equal type have equal probability, so

$$prob(01) = p(1-p) = (1-p)p = prob(10).$$

Therefore Chester has output 0 or 1 with equal probability; a perfectly random bit. If the parity is 0, then Chester knows he has drawn "00" or "11." He can do nothing with these bits so he discards them and draws two more.

Von Neumanns quantum protocol operates similarly. The explanation here draws heavily from [8]. Alice and Bob read in two identical states $|\psi\rangle = \sqrt{p}\,|0_A 0_B\rangle + \sqrt{1-p}\,|1_A 1_B\rangle$ from their source. The total state of the system at this point is

$$\begin{aligned} |\psi\rangle \otimes |\psi\rangle &= p\,|0_A 0_B\rangle\,|0_A 0_B\rangle + \sqrt{p(1-p)}\,|0_A 0_B\rangle\,|1_A 1_B\rangle \\ &\quad + \sqrt{p(1-p)}\,|1_A 1_B\rangle\,|0_A 0_B\rangle + (1-p)\,|1_A 1_B\rangle\,|1_A 1_B\rangle \\ &= p\,|00\rangle_A\,|00\rangle_B + \sqrt{p(1-p)}\,|01\rangle_A\,|01\rangle_B \\ &\quad + \sqrt{p(1-p)}\,|10\rangle_A\,|10\rangle_B + (1-p)\,|11\rangle_A\,|11\rangle_B\,. \end{aligned} \tag{66}$$

From here, Alice and Bob each flip their low qubits if their high qubits are $|\psi\rangle$ (a controlled not or CNOT operation). This operation sets the low qubit to the parity of the input.

$$\begin{aligned} |\psi\rangle \otimes |\psi\rangle &\rightarrow p\,|00\rangle_A\,|00\rangle_B + \sqrt{p(1-p)}\,|01\rangle_A\,|01\rangle_B \\ &\quad + \sqrt{(1-p)p}\,|11\rangle_A\,|11\rangle_B + (1-p)\,|10\rangle_A\,|10\rangle_B \\ &= \Big[p\,|0_A 0_B\rangle + (1-p)\,|1_A 1_B\rangle\Big]\,|0_A 0_B\rangle \\ &\quad + \sqrt{p(1-p)}\Big[\,|0_A 0_B\rangle + |1_A 1_B\rangle\Big]\,|1_A 1_B\rangle\,. \end{aligned} \tag{67}$$

Conditional on their second qubits being $|1\rangle$, their first qubits form an EPR pair $|\Phi^+\rangle$, which they output. Otherwise, their first qubits occupy a junk state

$$|\psi\rangle_{fail} = \frac{p}{\sqrt{p^2+(1-p)^2}}\,|0_A 0_B\rangle + \frac{1-p}{\sqrt{p^2+(1-p)^2}}\,|1_A 1_B\rangle$$

with which Alice and Bob can do nothing. They discard the qubits without outputting anything and read in two more copies of $|\psi\rangle$. Note that by discarding $|\psi\rangle_{fail}$, Alice and Bob waste the substantial entanglement contained in it. Other algorithms will outperform von Neumann's protocol by utilizing or "recycling" as much of that entanglement as possible.

4.1.2 Elias's Block Protocol

Von Neumann's protocol utilized the fact that the probability of drawing a string s is invariant under permutations on s, if s came from an i.i.d. source, to extract random bits from pairs of non-random bits. Elias's block protocol is a generalization of Von Neumann's protocol from pairs of non-random bits to strings of length N.

There are $\binom{N}{T}$ strings in type class (N, T). Elias's block protocol starts by assigning each of these strings a unique index number α from 0 to $\binom{N}{T}$-1. Since the probability of drawing any string is equal, α is a uniformly random variable over the range 0 to $\binom{N}{T}$-1. The randomness of α is the essence of Elias's block protocol. The trick is converting our random *variable* to random *bits*.

Here is the strategy we employ: Write $\binom{N}{T}$ as a binary number. For each digit binary digit L of $\binom{N}{T}$ that is 1, create a bin of size 2^L. Find some way of indexing all the strings in type class (N, T). Map the first 2^{L_0} strings of (N, T) to the $L = L_0$ bin, the second 2^{L_1} strings to the $L = L_1$ bin, and so on. Within each bin L, assign each string a binary index number α_L (which has length L since there are 2^L elements in bin L). Given we are in bin L, the intrabin index α_L is a string of L random bits.

Lets illustrate this strategy with two examples. The first is the case of $N = 4, T = 1$, illustrated in Fig. 4a. The size of type class (4,1) is strings

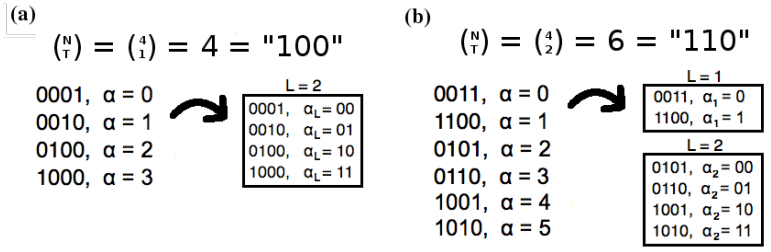

Figure 4: Illustration of Elias's block protocol for $N = 4$ and (a) $T = 1$, (b) $T = 2$. This diagram shows how we first map all strings into blocks by type class and then index within those blocks, outputting the index as a sequence of random bits. Notice the similarity of this entanglement concentration algorithm to the Schumacher compression algorithm (see Figure 2), although we use a different indexing scheme.

of is 4. Because the binary expansion of 4 has only one nonzero term, 2^2, we only need one bin, indexed by two bits. In other words, by taking the binary expansion of α we immediately get two random bits.

The second is the case of $N = 4, T = 2$, illustrated in Fig. 4b. In this example, we cannot simply convert α to binary and output it as random bits, because the size of type class (4,2) is $\binom{4}{2} = 6$, which is not a power of 2. However, we can represent 6 as $110 = 2^2 + 2^1$, and divide up the strings in type class (4,2) into two bins indexed by one and two bits respectively. The particular indexing scheme we use is irrelevant as long as we are consistent, although we will impose limits on it for the streaming protocol. It is not equally likely we will find ourselves in either bin. However, the intrabin indices are random bits. If we are in the first bin we output one random bit and in the second we output two. In the quantum case, if L is in a superposition of different *values*, then the *number* of random bits will also be in a superposition. Note that even

though the size of type class (4,2) is greater than the size of type class (4,1), it is possible to output fewer random bits for a length-4 string of type 2 than a length-4 string of type 1, depending on the way we bin our inputs.

As stated in the beginning of this section, Von Neumann's protocol is Elias's block protocol for $N = 2$ bits. If we find ourselves in $\binom{N}{T} = \binom{2}{0} = \binom{2}{2} = 1$, then we have one bin of size $2^0 = 1$ with 0 bits indexing it. Therefore we output nothing for $\alpha_L = \alpha_0$. If we have $\binom{N}{T} = \binom{2}{1} = 2$, then we have a bin of size $2^1 = 2$ with 1 bit indexing it. Therefore we output one random bit. In the earlier explanation, we used a CNOT targetting the low qubit as our indexing scheme, thus outputting "0" for "01" and "1" for "10," but our choice was arbitrary; by targetting the high qubit and outputting the lower, we could have implemented the opposite.

In the quantum implementation of Elias's block protocol, we start with a string of N qubits. We store N in some quantum register and set another quantum register to T. We then run our algorithm, determining both L and α_L. L we must store in another quantum register, while α_L we output. α_L is a variable length string of qubits in state $\rho = \frac{I}{2}$ with total length L. A quantum implementation of Elias's block protocol maps a set of N input qubits to an internal memory state (N, T, L) and outputs a string of random qubits α_L. Since both Alice and Bob perform the protocol and have perfectly correlated inputs and outputs because they perform their operations in the Schmidt basis, their α_Ls together form a string of EPR pairs. Blume-Kohout et. al. show that this protocol concentrates entanglement at a maximal rate (≈ 1 EPR pair per bit of input entanglement) as $N \to \infty$.

4.1.3 Serializing Elias's Protocol

Elias's block protocol operates on a sizeable chunk of qubits and requires very qubit-intensive calculation of large numbers like $\binom{N}{T}$. Furthermore, we cannot get any output EPR pairs until the entire algorithm finishes running. If we wish to extract additional EPR pairs afterwards, we must start the algorithm over from scratch–wasting potentially salvageable entanglement if $\binom{N}{T}$ at protocol termination was not a power of two.

We can ameliorate or entirely circumvent these problems by running Elias's protocol as a streaming algorithm [8]. We can consider the internal state of our quantum computer as representing a particular location on Pascal's triangle (see Fig. 1). Elias's protocol can be thought of as traversing the triangle, as we update N, the total number of qubits, and T, the type of our string (and L, as we output bits). Increasing N corresponds to moving down one row. We branch to the right when T increases with N, and we branch to the left when N increases without T. This corresponds to drawing a 1 or 0, respectively, from our input source.

The basic way the algorithm operates is by detecting when we could have reached a specific node (N, T, L) from two nodes ($N-1$, T_{0_0}, L_{0_0}) and ($N-1$, T_{0_1}, L_{0_1}) with equal probability, with a particular bit b denoting which "path" we took to reach (N, T, L). We then output b and update L to $L+1$, since

L indicates not only the bin which our total input string occupies but also the number of random bits we have output so far. Note that this sets limits upon the particular way we order strings within a bin; if 10010 corresponds to the $L = 3$ bin of ($N = 5$, $T = 2$), then 100100 must correspond to a bin $L >= 3$ at node ($N = 6$, $T = 2$), as it is impossible to "take back" bits b that we have already output. While the bits we output will not always be parts of our input string itself, we will end up outputting a substantial portion of our input string through the process of running this algorithm.

Given the importance of L, Fig. 1 is not quite suitable as a diagram for how we traverse Pascal's triangle. Instead we must separate each (N,T) into a set of nodes with all different possible values of L (dependent on the binary expansion of $\binom{N}{T}$, so between 0 and $\log N$ nodes): Once again, the streaming

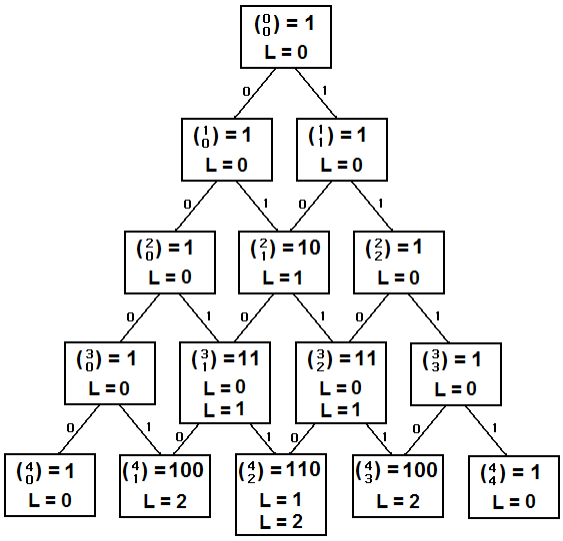

Figure 5: This diagram shows the basics of how sequentially running Elias's protocol corresponds to traversing Pascal's triangle. From any particular node, the user travels down the triangle by drawing bits b and branching left when $b = 0$ and right when $b = 1$, updating N, T, and L in the process.

implementation of Elias's protocol, developed in both its classical and quantum forms by Blume-Kohout et. al. [8], can be thought of as a set of rules for traversing this triangle.

The rules for N and T are simple. We update N by adding one for every new bit. We update T by adding one if our new bit is one.

L is more complicated because it represents the number of random bits we have output *and also* labels the bin containing the string we have drawn so far. In our discussion of Elias's block protocol we showed how the bins corresponded to digits in the binary expansion of $\binom{N}{T}$. L thus can be seen as a digit in a

binary number. Recall that by eq. 2

$$\binom{N}{T} = \binom{N-1}{T-1} + \binom{N-1}{T}.$$

We update L by a process exactly analogous to binary addition. We send $L = L_0$ to $L = L_0 + 1$ whenever the addition of eq. 2 involves adding two 1s in the L_0th digit. In other words, we update L whenever a "carry bit" is produced in the 2^Ls place of the addition. For notational convenience we define

$$\binom{N}{T}_L = \text{the } L\text{th digit of } \binom{N}{T}. \tag{68}$$

This can happen in one of three ways. Case 1 (Figure 6) is where we have $\binom{N-1}{T}_{L_0} = \binom{N-1}{T-1}_{L_0} = 1$, and no carry bits coming in from lower digits of the addition. In this case, we know we have two nodes leading into $\binom{N}{T}$, and we

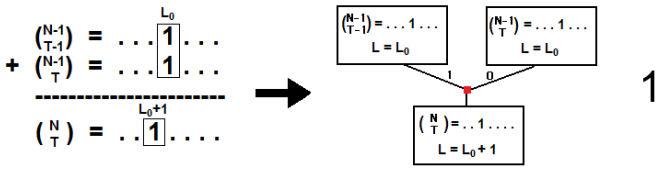

Figure 6: Case 1. Producing a carry bit in binary addition corresponds to emitting a random bit (denoted by red square) in Elias's streaming protocol. $\binom{N-1}{T}_{L_0} = \binom{N-1}{T-1}_{L_0} = 1$ corresponds to two $L = L_0$ bins at $\binom{N-1}{T}$ and $\binom{N-1}{T-1}$ from which we are equally likely to reach $\binom{N}{T}$.

know by the permutation invariance of types we are equally likely to have come from either one by drawing $b = 0$ or $b = 1$ respectively. Therefore, we can output b as a perfectly random bit.

In case 2 (Figure 7) we have $\binom{N-1}{T}_{L_0} = \binom{N-1}{T-1}_{L_0} = 1$ *AND* there is a carry bit coming in from a lower digit of the addition. Although we end up with a carry bit in the L_0th place while doing the addition, it never combines with any other bit; it just becomes a new bin at L_0 in addition to the new bin we find ourselves in at $L_0 + 1$. Therefore we can still output b as a perfectly random bit.

Case 3 arises when $\binom{N-1}{T}_{L_0} + \binom{N-1}{T-1}_{L_0} = 1$, i.e. when only one of them is 1, but there is a carry bit coming in from the $(L_0 - 1)$st digit. In these cases, we can definitely output b for the addition in the $(L_0 - 1)$st place, but if that carry bit is going to combine with the one in the L_0th place, what do we output? We only have one input bit b to output, and we've already used it.

Notice that the bit we output if we are coming from the L_0th place constrains our choice. From the perspective of the L_0th place, we have already outputted

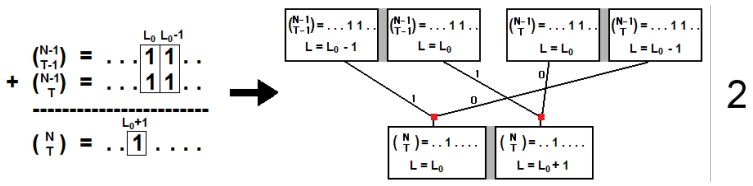

Figure 7: Case 2. Another situation in which we output random bits in correspondence with producing carry bits in binary addition. If we have carry bits coming from both the $(L_0 - 1)$st and L_0th places, they both create new bins at the L_0th and $(L_0 + 1)$st places without interfering with each other.

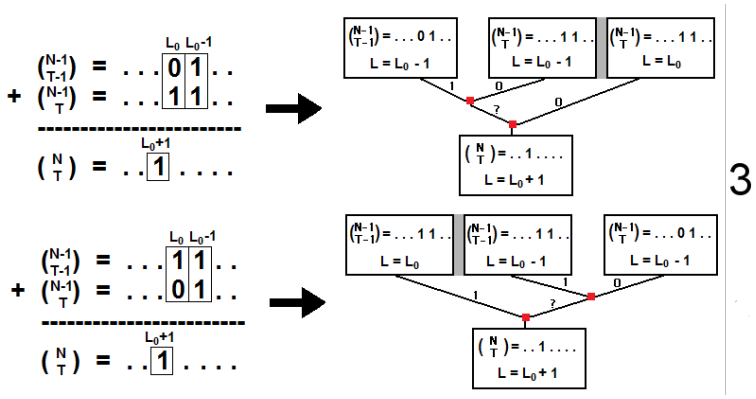

Figure 8: Carry bit case 3. Even if we don't have two 1s in the L_0th place, we can still produce a carry bit at $(L_0 + 1)$ if there is another carry bit coming up from the $(L_0 - 1)$st place.

L_0 random bits, and we're about to output b. To make b random, we must output its opposite on the carry path from the $L_0 - 1$st place. For the top subcase in Fig. 8, this means the bit denoted "?" must be 1, and for the bottom it must be 0. In both cases, the only bit that satisfies these criterion is $\binom{N-1}{T}_{L_0-1}$. Therefore we output $\binom{N-1}{T}_{L_0-1}$ instead of an input bit, if we are forced to output a bit in a place we have reached via a carry path.

How do we detect these three cases? If the result of our addition, $\binom{N}{T}$ (which we can calculate before updating L), ends up with 0 in its L_0th digit, that tells us we had to add exactly two 1s in the L_0th place. This covers case 1 in which both $\binom{N-1}{T-1}_{L_0}$ and $\binom{N-1}{T}_{L_0} = 1$ with no incoming carry, as well as case 3 in which only one of them is 1 but there is an incoming carry bit. However, it does not cover case 2, in which both $\binom{N-1}{T-1}_{L_0} = \binom{N-1}{T}_{L_0} = 1$ and there is an incoming carry bit from a lower digit. In this case, the result of the addition will have 1 in the L_0th place even though there were at least two ways of reaching

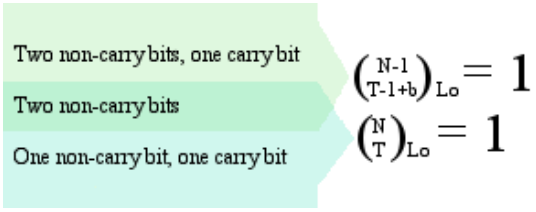

Figure 9: Graphical illustration of how boolean conditions $\binom{N-1}{T-1+b}_{L_0} = 1$ and $\binom{N}{T}_{L_0} = 0$ together fully cover the three cases (illustrated in figs. 6, 7, and 8) under which we must update L and output a random bit.

(N, T, L). We need another condition to check for case 2.

Consider what happens when $\binom{N-1}{T-1+b}_{L_0} = 1$. Although in the quantum case we are working with superpositions, we can say that we reached our current state (N, T) from $(N-1, T-b)$. The state from which we did *not* reach (N, T) is thus $(N-1, T-(1-b)) = (N-1, T-1+b)$. We know simply by virtue of the fact that we arrived at state (N, T, L_0) that $(N-1, T-b)$ has 1 in its L_0th digit. If $(N-1, T-1+b)$ also has 1 in its L_0th digit, that means there are two non-carry bits and possibly a third carry bit in our addition. We must update L and output b whenever $\binom{N-1}{T-1+b}_{L_0} = 1$ or $\binom{N}{T}_{L_0} = 0$.

After outputting b, we must check to see if it will combine with any 1s in higher digits, which requires a loop over the remaining digits. What is the loop condition? If we have just emitted a carry bit at $L = L_0$, and we find that the $L_0 + 1$st bits of $\binom{N-1}{T-1}$ and $\binom{N-1}{T}$ are the same, then we should stop looping (as that indicates one of the first two cases discussed in this section or a case in which both are 0). However, if there is only one non-carry bit (i.e. if $\binom{N-1}{T-1}_{L_0+1}$ and $\binom{N-1}{T}_{L_0+1}$ are different), then we must fuse with it and output $\binom{N-1}{T}_{L_0}$ a new random bit. Then we must check the loop condition again.

We can now condense this discussion into a concise statement of Blume-Kohout et. al.'s streaming version of Elias's protocol:

Elias's Streaming Protocol

1. `WHILE( input stream not empty ) DO`
2. {
3. `Read a bit` b `from input stream.`
4. `Update` $N \to N+1$ `and` $T \to T+b$.
5. `IF (` $\binom{N}{T} = 0$ `or` $\binom{N-1}{T-1+b}_L = 1$ `)`
6. {

7. `output` b `and set` $L \to L+1$.

8. `WHILE (` $\binom{N-1}{T}_L \neq \binom{N-1}{T-1}_L$ `)`

9. `output` $\binom{N-1}{T}_L$ `and set` $L \to L+1$.

10. }

11. }

as well as Figure 10, an updated version of Figure 5 that illustrates the paths we must take to get from node to node.

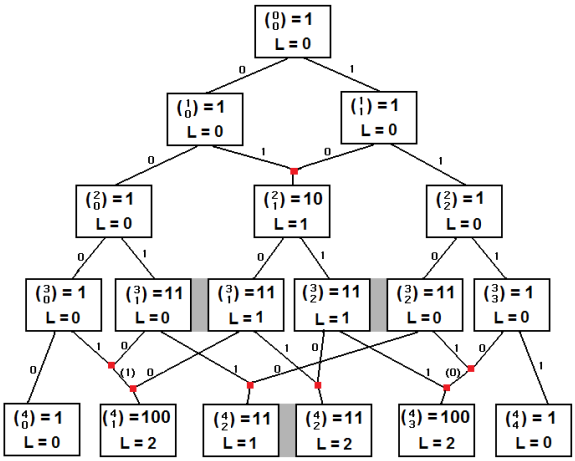

Figure 10: Blume-Kohout et. al.'s implementation of Elias's streaming protocol from $N = 0$ to $N = 4$. Starting at $N = 0, T = 0$ at the top of the diagram, we draw bits b and move downwards, taking the left path if $b = 0$ and the right path if $b = 1$. Whenever we reach a red square, we output the bit corresponding to our current path. This output bit is always b unless parenthesized, in which case it is $\binom{N-1}{T}_L$. Assuming we draw b from an I.I.D. source, our output bit will be perfectly random in the classical case and completely entangled with its distant pair in the quantum case.

In terms of physical implementation, all we need is a coherent variable length output tape, three quantum registers for N, T, and L each of size $O(\log N)$, and other apparatus capable of performing basic logical operations and calculating specific bits (not the entirety) of $\binom{N}{T}$. Blume-Kohout et. al. discuss the requirements in detail, but it suffices to say here that this streaming version of Elias's protocol can be implemented with current or near-future technology.

There are limitations on how we can use this algorithm that Blume-Kohout et. al. do not fully comment on. The way Elias's streaming protocol outputs data is by pushing EPR pairs onto a variable length output tape. However,

if Alice or Bob want an EPR pair, they cannot just grab one off the output tape without disrupting the algorithm's operation. Even if we implement the tape as a kind of stack, where we can push as many qubits as we like onto it and pop qubits off without learning the length of the tape, we nevertheless learn something when we pop a qubit off the tape: either we failed to pop a qubit, and the number of qubits on the tape is 0, or we succeeded, and the number of qubits on the tape is nonzero. Either of these outcomes will disrupt the protocol. Blume-Kohout et. al. discuss the implications of failing to pop a qubit, but they do not consider the similarly disruptive effects of success.

The number of qubits on the tape is L. The quantum register representing L must remain in a coherent superposition of different values so that paths from all nodes may interfere with one another to produce random results. As we get further and further into the triangle, each new bit's randomness depends on the equal likelihood of an increasing number of paths from all nodes, including some with $L = 0$. Once we pop an EPR pair, we learn that $L > 0$. As soon as we do that, the algorithm is doomed to fail, because all paths stemming from $L = 0$ bins can no longer interfere with others. For example, if we successfully pop an EPR pair off the tape at $N = 2$ (telling us $L = 1$ and our string is "01" or "10"), then when we reach $N = 4$, the string "0010" is no longer as likely as "0100." Those two paths cannot interfere to form new random bits at $\binom{N}{T} = \binom{4}{1} = 100$. But the algorithm still thinks they are. It reaches $\binom{N}{T} = 100$, sees that the $(L = 1)$st element is 0, and decides to output $b = 0$. But b is not a random bit because there is no other path outputting 1; the algorithm outputs an incorrect value.

If Alice or Bob try to pop off an EPR pair at any point, errors like this will eventually crop up. If they measure at $N = 2$, the first potential error occurs at $N = 4$. If they measure at $N = 4$, the first potential error occurs at $N = 7$. As N gets very large, the probable value of L increases (so the disruption created by measurement decreases), and it will most likely take longer and longer for the first potential error to appear (and perhaps longer still for those errors to become likely). Nevertheless, errors will definitely occur if Alice and Bob do anything that gives them information about L, the length of the output tape. Furthermore, even if Alice does nothing wrong, Bob's mistaken measurement will disrupt Alice's results as well. Blume-Kohout et. al. recognize the danger in looking for an EPR pair and finding none, so they suggest an "incubator" strategy to avoid problems by always making sure there are at least a few qubits on the output tape before popping one off. However, this only exacerbates the problem; if Alice or Bob must pop qubits off the output tape, they definitely don't want to learn that there are at least a few; they only want to learn that there is at least one.

However, despite these limitations, Alice and Bob can use Elias's streaming protocol to transform sets of identical and partially entangled qubits into EPR pairs using only local operations, provided they work in the Schmidt basis. The streaming Elias protocol presented here and in [8] is effectively a classical algorithm made universally quantum by our ability to estimate the Schmidt basis with trivial loss of entanglement.

4.2 3 Qubit Entanglement Concentration

Let Alice, Bob, and Chester each have one qubit. If each of those qubits are in pure states $|A\rangle$, $|B\rangle$, and $|C\rangle$ respectively, then the joint state of the system $|\psi\rangle_{abc}$ is just $|A\rangle \otimes |B\rangle \otimes |C\rangle$ (product state) and measuring any of their qubits will not affect the others. There is no entanglement between the qubits of Alice, Bob, and Chester.

In general there is entanglement between (two or more of) the three qubits whenever this is not the case. However, there are multiple types of such entanglement. It is possible Alice and Bobs qubits are entangled but only with each other, and not with Chesters. There are three types of such bipartite entanglement for three qubits,

$$|A\rangle \otimes \Big(|B_1\rangle \otimes |C_1\rangle + |B_2\rangle \otimes |C_2\rangle \Big), \tag{69}$$

$$|B\rangle \otimes \Big(|A_1\rangle \otimes |C_1\rangle + |A_2\rangle \otimes |C_2\rangle \Big), \tag{70}$$

$$|C\rangle \otimes \Big(|A_1\rangle \otimes |B_1\rangle + |A_2\rangle \otimes |B_2\rangle \Big). \tag{71}$$

In Eqs. 69, 70, 71, measuring Alice's, Bob's, and Chester's qubits respectively will have no effect on anyone else's qubits; but measuring either one of the two remaining qubits will affect the third.

The most general situation occurs when all three qubits have some sort of entanglement relation, i.e. when any measurement of any party's qubit will affect both of the others. As Dur, Vidal, and Cirac show [12], there are two inequivalent types of tripartite entanglement.

Two multipartite states are said to have inequivalent entanglement if they cannot be converted into one another by local operations and classical communication (LOCC), even if we allow that conversion to have a chance of failure, i.e. make those operations stochastic (SLOCC). States that can be converted into one another by SLOCC are said to be SLOCC equivalent. In formulating and universalizing Elias's protocol, we relied upon the fact that all states of two qubits with nonzero entanglement are SLOCC equivalent to EPR pairs. This followed from special properties of the Schmidt decomposition of two qubit states, which do not hold for three qubits.

In simple terms, some states of three qubits are representable as a superposition of two product states, while others can only be represented as a superposition of three or more. Since the minimal number of product terms for any given state is invariant under SLOCC [12], three-qubit states that must be written as a sum of at least three product states cannot be converted with any chance of success into states that can be written as a sum of just two.

Within each of these categories, there are particular states that [12] labels $|GHZ\rangle$ and $|W\rangle$ that may be thought of as representative:

$$|GHZ\rangle = \frac{|000\rangle + |111\rangle}{\sqrt{2}} \tag{72}$$

and

$$|W\rangle = \frac{|001\rangle + |010\rangle + |100\rangle}{\sqrt{3}} \tag{73}$$

If we divide state $|GHZ\rangle$ into any two subsystems, regardless which subsystems we choose, they will be maximally entangled. State $|GHZ\rangle$ is also maximally entangled with respect to the n-tangle, a commonly used entanglement measure for states of many qubits. Subsystems of state $|W\rangle$ are not maximally entangled with each other, but two-qubit subsystems of $|W\rangle$ retain a maximal amount of entanglement between their qubits when the other qubit of $|W\rangle$ is measured (inadvertently or otherwise). One application of $|W\rangle$ states is discussed in Appendix B.

In the upcoming sections we will assume Alice, Bob, and Chester have a source that outputs identical three-qubit states of known entanglement class (GHZ in the next section, W in the one after). They each receive one qubit of the three and can perform only local operations (perhaps with classical communication) on them. The question of whether and how they can concentrate the states they receive to GHZ or W states will be the subject of the rest of the paper.

4.2.1 Elias's Protocol and GHZ States

The most general form of a GHZ-class 3-qubit state is

$$|\psi\rangle = \sqrt{p}\,|a_1\rangle\,|b_1\rangle\,|c_1\rangle + \sqrt{1-p}\,|a_2\rangle\,|b_2\rangle\,|c_2\rangle\,, \tag{74}$$

where Alice, Bob, and Chester respectively possess the state's first, second, and third qubits. $|a_1\rangle$ are $|a_2\rangle$ not in general orthogonal, and similar for b and c. This is a consequence of the result that there is no analogue of the Schmidt basis for three qubits [12]. While we can write down

$$|\psi\rangle = \lambda_1\,|a_1'\rangle\,\left|\phi_1^{bc}\right\rangle + \lambda_2\,|a_2'\rangle\,\left|\phi_2^{bc}\right\rangle\,, \tag{75}$$

where $|a_i'\rangle$ are orthogonal eigenvectors of ρ_A and ϕ_i^{bc} are orthogonal eigenvectors of $\rho_B C$, we cannot in general further subdivide this such that $|a_i\rangle\,, |b_i\rangle\,,$ and $|c_i\rangle$ in eq. 74 are all orthogonal pairs of vectors.

For now we will assume Alice, Bob, and Chester possess many identical copies of a state

$$|\psi\rangle = \sqrt{p}\,|000\rangle + \sqrt{1-p}\,|111\rangle\,.$$

We will briefly show that Von Neumann's and Elias's protocols are a means of transforming many copies of $|\psi\rangle$ to $|GHZ\rangle$s (72). Afterwards, we will discuss the possibility of converting our general GHZ-class entangled state to this more specific form.

Assume Alice, Bob, and Chester draw two copies of $|\psi\rangle = \sqrt{p}\,|000\rangle + \sqrt{1-p}\,|111\rangle$:

$$\begin{aligned}|\psi\rangle\otimes|\psi\rangle &= p\,|0_A0_B0_C\rangle\,|0_A0_B0_C\rangle + \sqrt{p(1-p)}\,|0_A0_B0_C\rangle\,|1_A1_B1_C\rangle \\ &\quad +\sqrt{p(1-p)}\,|1_A1_B1_C\rangle\,|0_A0_B0_C\rangle + (1-p)\,|1_A1_B1_C\rangle\,|1_A1_B1_C\rangle \\ &= p\,|00\rangle_A\,|00\rangle_B\,|00\rangle_C + \sqrt{p(1-p)}\,|01\rangle_A\,|01\rangle_B\,|01\rangle_C \\ &\quad +\sqrt{p(1-p)}\,|10\rangle_A\,|10\rangle_B\,|10\rangle_C + (1-p)\,|11\rangle_A\,|11\rangle_B\,|11\rangle_C\,. \quad (76)\end{aligned}$$

Now each party performs a CNOT operation, with the second bit as target:

$$\begin{aligned}|\psi\rangle\otimes|\psi\rangle &\rightarrow p\,|00\rangle_A\,|00\rangle_B\,|00\rangle_C + \sqrt{p(1-p)}\,|01\rangle_A\,|01\rangle_B\,|01\rangle_C \\ &\quad +\sqrt{p(1-p)}p\,|11\rangle_A\,|11\rangle_B\,|11\rangle_C + (1-p)\,|10\rangle_A\,|10\rangle_B\,|10\rangle_C \\ &= \Big[p\,|0_A0_B0_C\rangle + (1-p)\,|1_A1_B1_C\rangle\Big]\,|0_A0_B0_C\rangle \\ &\quad +\sqrt{p(1-p)}\Big[\,|0_A0_B0_C\rangle + |1_A1_B1_C\rangle\Big]\,|1_A1_B1_C\rangle\,. \quad (77)\end{aligned}$$

Conditional on Alice, Bob, and Chester's second qubits each being $|1\rangle$, their first qubits together form $|GHZ\rangle$.

If we trace out any two qubits from $|\psi\rangle$, we find the third is in a state $p\,|0\rangle\langle 0| + (1-p)\,|1\rangle\langle 1|$, which denotes an i.i.d. source. All of the same assumptions we made in applying Elias's protocol for two qubit entanglement (effective i.i.d. sources that are perfectly correlated) thus hold for the three qubits of $|\psi\rangle$. Furthermore we have shown quite explicitly that Von Neumanns protocol, the $N = 2$ case of Elias's protocol, works for $|\psi\rangle$. Therefore, by the arguments of the previous section, we can conclude that Elias's protocol (optimally) transforms states $|\psi\rangle$ into $|GHZ\rangle$.

However, because $|a_1\rangle$ and $|a_2\rangle$ are not necessarily orthogonal, and similar for b and c, there is no way for Alice, Bob, and Chester to reliably express

$$\sqrt{p}\,|a_1\rangle\,|b_1\rangle\,|c_1\rangle + \sqrt{1-p}\,|a_2\rangle\,|b_2\rangle\,|c_2\rangle$$

as

$$\sqrt{p}\,|000\rangle + \sqrt{1-p}\,|111\rangle\,.$$

Since our three parties are separated in space, the only changes of basis they may apply are those represented by local unitary operators $U_A \otimes U_B \otimes U_C$. No transformation of this form can make $|a_1\rangle$ orthogonal to $|a_2\rangle$, or even change the angle between them in Hilbert space. If we allow non-unitary operations, say all invertible local operators or SLOCC, then we can transform $\sqrt{p}\,|a_1\rangle\,|b_1\rangle\,|c_1\rangle + \sqrt{1-p}\,|a_2\rangle\,|b_2\rangle\,|c_2\rangle$ to $\sqrt{p}\,|000\rangle + \sqrt{1-p}\,|111\rangle$ or even $|GHZ\rangle$, but not with unit probability [12]. Furthermore, it is not possible to do this without some knowledge of a, b, and c, which may require entanglement-destroying

measurement. Whereas in the two-qubit case we could always transform our qubits to the Schmidt basis with negligible loss of entanglement, in the three-qubit case there is no Schmidt basis that can be made common for all three qubits by simple entanglement-preserving unitaries. This puts a fundamental restriction on our ability to apply Elias's protocol to GHZ-class three-qubit states.

4.2.2 W States

W states are states $|\psi\rangle$ of three qubits that cannot be expressed as a sum of two product terms as in eq. 74. In general we can express them as a sum of three product terms, but following [12] we can always express them in the following useful form:

$$|\psi\rangle = \sqrt{a}\,|0_A 0_B 1_C\rangle + \sqrt{b}\,|0_A 1_B 0_C\rangle + \sqrt{c}\,|1_A 0_B 0_C\rangle + \sqrt{d}\,|0_A 0_B 0_C\rangle\,. \tag{78}$$

The goal of an entanglement concentration protocol would be to concentrate states of this form to states $|W\rangle$ (73). Because Alice's, Bob's, and Chester's qubits will never be perfectly correlated, adapting an algorithm like Elias's protocol to concentration of $|W\rangle$ states does not seem viable.

However, we can use the fact that if Alice sees a $|1\rangle$, then Bob and Chester will most certainly see $|0\rangle$ to move closer towards a potential algorithm. Let Alice, Bob, and Chester each draw three copies of $|\psi\rangle$, so the total state of the system is $|\psi\rangle \otimes |\psi\rangle \otimes |\psi\rangle$. Have each of them set their third qubit to the parity of all three qubits, which is $|1\rangle$ when one or three of their qubits is $|1\rangle$. If each party measures their third qubit to be $|1\rangle$, after this process, then we know that Alice, Bob, and Chester each have exactly one qubit which is $|1\rangle$ and two qubits which are $|0\rangle$. Furthermore, the amplitude of all such states is $\sqrt{abc}$, so they are all equally likely. The state of the system will thus collapse to

$$\begin{aligned}
|\psi\rangle \;\rightarrow\; & \tfrac{1}{27}\Big[\; |00\rangle_A\,|01\rangle_B\,|10\rangle_C + |00\rangle_A\,|10\rangle_B\,|01\rangle_C \\
& + |01\rangle_A\,|00\rangle_B\,|10\rangle_C + |01\rangle_A\,|10\rangle_B\,|00\rangle_C \\
& + |10\rangle_A\,|00\rangle_B\,|01\rangle_C + |10\rangle_A\,|01\rangle_B\,|00\rangle_C \Big]\,|1_A 1_B 1_C\rangle \qquad (79)\\
= \; & \tfrac{1}{27}\Big[\; |0_A 0_B 1_C\rangle\,|0_A 1_B 0_C\rangle + |0_A 0_B 1_C\rangle\,|1_A 0_B 0_C\rangle \\
& + |0_A 1_B 0_C\rangle\,|0_A 0_B 1_C\rangle + |0_A 1_B 0_C\rangle\,|1_A 0_B 0_C\rangle \\
& + |1_A 0_B 0_C\rangle\,|0_A 0_B 1_C\rangle + |1_A 0_B 0_C\rangle\,|0_A 1_B 0_C\rangle \Big]\,|1_A 1_B 1_C\rangle \\
= \; & \tfrac{1}{27}\Big[\; 3\,|W\rangle \otimes |W\rangle - |0_A 0_B 1_C\rangle\,|0_A 0_B 1_C\rangle \\
& - |0_A 1_B 0_C\rangle\,|0_A 1_B 0_C\rangle - |1_A 0_B 0_C\rangle\,|1_A 0_B 0_C\rangle \Big]\,|1_A 1_B 1_C\rangle
\end{aligned}$$

This state is appealing because it is very close to $|W\rangle \otimes |W\rangle \otimes |111\rangle$ and shares many of its properties; the chance of Alice, Bob, or Chester measuring their first or second qubits to be $|1\rangle$ is $\frac{1}{3}$, and any time one of them measures $|1\rangle$ in their first or second qubit, the others measure $|0\rangle$ in their corresponding qubits. Unfortunately, all three parties' first qubits are entangled with their second qubits. Eq. 79 is not a separable state; although we want corresponding qubits between parties to be dependent, we need any two qubits belonging to one party to be independent. This is not the case.

One would hope that, through some sort of additional work, perhaps involving additional measurements, we might be able to extract at least one $|W\rangle$ state from eq. 79. After all, it took two qubits for von Neumann's protocol to extract one EPR pair, so why not three qubits for this protocol to extract one copy of $|W\rangle$? However, what we hope this section has emphasized is that the mathematical structure of entangled three-qubit messages is very different from that of two-qubit messages, and many concepts we relied on for two qubits, such as the Schmidt basis (not just decomposition), have no three-qubit generalization. If a protocol for extracting $|W\rangle$ states eq. 79 from does exist, it may not have any resemblance to protocols we used for two qubits.

5 Conclusion

In this paper we have considered a number of message sources, described them mathematically, and have presented algorithms to compress and concentrate the information and entanglement contained in their messages. Systems we have considered include i.i.d. sources of bits, i.i.d. sources of qubits, configurations of fermionic and bosonic grand canonical ensembles, and entangled pairs and triplets of qubits divided among distant actors.

In Section 2 we gave an overview of a simple compression algorithm, meant to compress bits and qubits from i.i.d. sources, that directly follows in the classical case from Shannon's coding theorem [10] and in the quantum case from Schumacher's [9]. We presented and explained Cleve and DiVincenzo's classical and quantum implementations of this algorithm [2].

In Section 3 we considered data obtained by measuring configurations of fermions and bosons and found that it was independent but not identically distributed, with the distribution given by the Fermi-Dirac distribution for fermions and a geometric distribution for bosons. We were able to meaningfully calculate an entropy for each system that described the amount of information in bits observers gain from measuring their configurations. We cited Johnson and Suhov's proven result [3] that Lempel-Ziv compression (see Appendix A) is an appropriate compression algorithm for such systems and commented on alternatives.

In Section 4 we presented Blume-Kohout et. al.'s algorithm, based on Elias's randomness extraction protocol, for Alice and Bob to locally concentrate entanglement from a shared set of partially entangled pairs of qubits into EPR pairs [8]. We pointed out a minor limitation on when Alice and Bob can access the

protocol's output. We also considered the problem of entanglement concentration for three qubits shared between three parties, and commented on limitations within which any potential concentration protocol would need to work.

There are still numerous sources of information that we do not know how to describe. For example, can we find some way of writing an aritrary quantum system defined by a Hamiltonian $\mathcal{H}$ as a message source, and calculate its entropy? And if so, what sorts of insights would an information-theoretic understanding of such physical systems impart about them? What sorts of insights could it impart about the universe?

Some final thoughts: initial objections to quantum mechanics claimed that a theory involving things like probability amplitudes and physical systems existing in superpositions of different states (sometimes even superpositions of existence and nonexistence) was somehow fundamentally unsatisfying. Rather than actual superposition between two states, there must be some sort of hidden variable that describes the "true" state, objectors insisted. This is more or less Einstein, Podolsky, and Rosen's argument, in their 1935 paper [1] that introduced ideas of entanglement and even named the EPR pair upon which we draw so heavily in this document. Hidden variable theories were eventually disproven, with the alternative view of reality being one based on entanglement. We can understand this paradigm shift from an information-theoretic perspective. If there was going to be a variable, hidden or not, describing the state of a system, it would need to be stored somewhere. Entanglement as a quantity describes how much we learn about one system when we measure another. Thus for information about the state of one physical system to exist, it must be localized in a second physical system that is entangled with the first. Hidden variable theory assumes that information about a particle's state, i.e. a variable, exists independently of other physical systems, but the entanglement-based view of information we have presented in this paper fundamentally invalidates that assumption. All information is stored in physical systems.

This perspective can give us insight into the nature of measurement. The process of measurement in quantum mechanics is often represented as a discrete event: an experimenter "performs a measurement" to discover the state of the system, and his/her conscious knowledge of the system's state coincides with that state's collapse into an eigenstate of the operator associated with the measurement. But the collapse of the wavefunction is not dependent on any individual's conscious knowledge. What then constitutes a measurement? Exactly when does the wavefunction collapse? We might guess it collapses when *information* about a system's state exists in the universe and is theoretically accessible to an observer. If this is true, then a quantum state is considered measured when it is entangled with its observer. States of physical systems become entangled with others via laws of interaction, which physics attempts to discover and describe. As Nielsen and Chuang argue ([7] p. 80), quantum mechanics is merely a mathematical framework in which to describe these laws. Measurement is simply the rapid spread of entanglement from one system to its observer, which is a consequence of physical laws governing interaction. If the entire universe could fit inside one qubit of an EPR pair, it would see another

qubit whose state was definite.

To gain a greater understanding of how information moves through the universe is to move towards a deeper understanding of physics in general. This paper has described the information-theoretic properties of a number of interesting systems, but it is important to recognize how much more there is to learn, and the scope of the knowledge we can gain from understanding information.

6 Acknowledgments

Thanks to Peter Love for guiding me, putting up with my silly antics, and teaching me an incredible amount about how to write, present, and do academic research. Thanks to Suzanne Amador Kane for academically advising me, and to the entire physics department (students and faculty) for creating a wonderful environment in which to learn.

A Classical Lempel-Ziv Compression

Lempel-Ziv is an umbrella term used to describe a wide variety of adaptive variable-length encoding schemes, i.e. compression algorithms that get better at compressing their input strings the further along they are in the encoding process. Lempel-Ziv algorithms were first introduced in [13], but since then many different variants have been written. In this section we will describe a specific case of Lempel-Ziv that is particularly helpful for understanding Lempel-Ziv algorithms in general, but keep in mind that there are many places where we could alter the rules slightly and still obtain both asymptotically optimal compression and an idiosyncratic Lempel-Ziviness of algorithm.

Johnson and Suhov [3] show that Lempel-Ziv is well suited to compressing data from particular physical systems we discuss in Section 3. In this Appendix we will just give an overview of a completely classical implementation of it.

Let us assume our input is the string

$$abaabaabaabaabaabaabaabaabaab.$$

We define a Lempel-Ziv compression algorithm as follows:

1. Initialize a dictionary (a data structure mapping strings to indices) with all characters in our alphabet mapped to the lowest indices.
2. Begin at the start of the input.
3. Starting at our current location on the input, find the longest dictionary match, and see how many times it repeats itself.
4. Add to the output the index or reference to that dictionary entry followed the number of times it was repeated.

5. Add a new entry to the dictionary, with index one higher than the most recent addition, that represents the input just matched concatenated with the symbol that broke the pattern.

6. Move our current location on the input forward to the symbol that broke the pattern, and repeat from step 3 until we reach the end of input.

Let's illustrate this process on our example input.

1. We initialize the dictionary to
   $a \to 0$
   $b \to 1$.

2. We begin at
   $$\mathbf{a}baabaabaabaabaabaabaabaabaab.$$
   - The longest dictionary match at current location is $a \to 0$. It repeats once.
   - Add $(0, 1)$; to the output: $(0, 1)$;
   - The symbol that broke the pattern was b. Add $ab \to 2$ to the dictionary.

3. Move to
   $$a\mathbf{b}aabaabaabaabaabaabaabaabaab.$$
   - The longest dictionary match at current location is $b \to 1$. It repeats once.
   - Add $(1, 1)$; to the output: $(0, 1); (1, 1)$;
   - The symbol that broke the pattern was a. Add $ba \to 3$ to the dictionary.

4. Move to
   $$ab\mathbf{a}abaabaabaabaabaabaabaabaab.$$
   - The longest dictionary match at current location is $a \to 0$. It repeats twice.
   - Add $(0, 2)$; to the output: $(0, 1); (1, 1); (0, 2)$;
   - The symbol that broke the pattern was b. Add $aab \to 4$ to the dictionary.

5. Move to
   $$abaa\mathbf{b}aabaabaabaabaabaabaabaab.$$
   - The longest dictionary match at current location is $ba \to 3$. It repeats once.
   - Add $(3, 1)$; to the output: $(0, 1); (1, 1); (0, 2); (3, 1)$;

- The symbol that broke the pattern was a. Add $baa \rightarrow 5$ to the dictionary.

6. Move to
$$abaaba\mathbf{a}baabaabaabaabaabaabaab.$$
   - The longest dictionary match at current location is $ab \rightarrow 2$. It repeats once.
   - Add $(2, 1)$; to the output: $(0, 1); (1, 1); (0, 2); (3, 1); (2, 1);$
   - The symbol that broke the pattern was a. Add $aba \rightarrow 6$ to the dictionary.

7. Move to
$$abaabaab\mathbf{a}abaabaabaabaabaabaab.$$
   - The longest dictionary match at current location is $aab \rightarrow 4$. It repeats seven times.
   - Add $(4, 7)$; to the output: $(0, 1); (1, 1); (0, 2); (3, 1); (2, 1); (4, 7);$
   - We reached the end of input, so we're done.

Final dictionary:
$a \rightarrow 0$
$b \rightarrow 1$
$ab \rightarrow 2$
$ba \rightarrow 3$
$aab \rightarrow 4$
$baa \rightarrow 5$
$aba \rightarrow 6$
Final output:
$(0, 1); (1, 1); (0, 2); (3, 1); (2, 1); (4, 7);$
Note that our output is a data structure, which makes a simple bitstring-to-bitstring style compression a little more complicated, and in many cases makes the output longer than the input. It is only in the case of asymptotically long strings that our compression algorithm gives us a "good" compression rate (i.e. to entropy).

Decompression is very similar to compression. If we now throw away most of our dictionary, it turns out that as long as we know $a \rightarrow 0$ and $b \rightarrow 1$ we can reconstruct our entire input from just our output. In this sense Lempel-Ziv has a fundamental reversibility, which ends up displaying a surprising symmetry.

The decompression algorithm is as follows:

1. Initialize a dictionary with all characters in our alphabet mapped to the lowest indices.

2. Begin at the first entry of the compressed input list.

3. Look up what dictionary entry is being referenced. Let's call it X. See how many times it repeats, and add that to the decompressed output.

4. Look ahead to the next entry of the compressed input list and see what dictionary entry it references. Let's call it Y.

5. The first letter of Y is what must have broke the pattern when we initially constructed the dictionary during compression. Concatenate X with the first letter of Y and add it to the dictionary.

6. Move our current location on the compressed input list to the next entry, and repeat from step 3.

Note the strong similarity of the decompression algorithm to the compression algorithm. Let's use the same example to illustrate decompression:

1. We initialize the dictionary to
   $a \to 0$
   $b \to 1$.

2. We begin at
   $$\mathbf{(0,1)}; (1,1); (0,2); (3,1); (2,1); (4,7);$$
   - Current entry $(0,1)$; implies $a \to 0$ repeats once.
   - Add a to the output: a
   - Next entry $(1,1)$; references $b \to 1$.
   - First letter of b is b. Add $ab \to 2$ to the dictionary.

3. Move to
   $$(0,1); \mathbf{(1,1)}; (0,2); (3,1); (2,1); (4,7);$$
   - Current entry $(1,1)$; implies $b \to 1$ repeats once.
   - Add b to the output: ab
   - Next entry $(0,2)$ references $a \to 0$.
   - First letter of a is a. Add $ba \to 3$ to the dictionary.

4. Move to
   $$(0,1); (1,1); \mathbf{(0,2)}; (3,1); (2,1); (4,7);$$
   - Current entry $(0,2)$ implies $a \to 0$ repeats twice.
   - Add aa to the output: $abaa$
   - Next entry $(3,1)$ references $ba \to 3$.
   - First letter of ba is b. Add $aab \to 4$ to the dictionary.

5. Move to
   $$(0,1); (1,1); (0,2); \mathbf{(3,1)}; (2,1); (4,7);$$
   - Current entry $(3,1)$ implies $ba \to 3$ repeats once.
   - Add ba to the output: $abaaba$
   - Next entry $(2,1)$ references $ab \to 2$.

- First letter of ab is a. Add $baa \rightarrow 5$ to the dictionary.

6. Move to
$$(0,1);(1,1);(0,2);(3,1);\mathbf{(2,1)};(4,7);$$
   - Current entry $(2,1)$ implies $ab \rightarrow 2$ repeats once.
   - Add ab to the output: $abaabaab$
   - Next entry $(4,7)$ references $aab \rightarrow 4$.
   - First letter of aab is a. Add $aba \rightarrow 6$ to the dictionary.

7. Move to
$$(0,1);(1,1);(0,2);(3,1);(2,1);\mathbf{(4,7)};$$
   - Current entry $(2,1)$ implies $aab \rightarrow 2$ repeats seven times.
   - Add $aabaabaabaabaabaabaab$ to the output: $abaabaabaabaabaabaabaabaabaab$
   - We reached the end of input, so we're done.

Final dictionary:
$a \rightarrow 0$
$b \rightarrow 1$
$ab \rightarrow 2$
$ba \rightarrow 3$
$aab \rightarrow 4$
$baa \rightarrow 5$
$aba \rightarrow 6$
Final output:
$abaabaabaabaabaabaabaabaabaab$
This rather laborious example hopefully makes clear the beautiful adaptability and the fearful symmetry of Lempel-Ziv. In both compression and decompression we construct the same dictionary and use it in a similar manner.

B W State Password Generation

This appendix briefly describes an application of $|W\rangle$ states to cryptography and password generation.

Assume Alice, Bob, and Chester each have N qubits, each of which forms a W state with corresponding qubits in possession of the other parties. Each also has a random string of N classical bits, a, b, and c respectively, that they privately generate. Alice, Bob, and Chester then begin to measure all of their qubits. When Alice measures qubit i, she measures it in the X basis if $a = 0$ and in the Z basis if $a = 1$. She records her result as 0 or 1, depending on the result of the measurement, and moves on to the next qubit. By doing so she creates a new string of result bits, s_a. Bob and Chester do the same, using strings b and c respectively, to produce strings s_b and s_c.

After all qubits have been measured, Alice, Bob, and Chester send their classical bit strings a, b, and c to one another. They then save all qubit measurement results in their strings s for which $a = b = c$. They throw away every other entry of s. At the end of this process, they will be left with strings of approximately $\frac{N}{8}$ bits. At any index in these strings, two of those strings will be 0 and the third will be 1. If any two of Alice, Bob, and Chester ever compare these strings of qubits, they will have some probability of seeing 01, some probability of seeing 00, but no probability of ever seeing 11.

Alice, Bob, and Chester can use this as a password system to ensure private communication. To access some piece of private information, a party must submit a long password string that does not need to match Alice, Bob, or Chester's strings exactly, but must be compatible with it. In other words, when Alice, Bob, or Chester compare the other party's string to theirs, there must never be any instance in which both strings are 1 at the same index.

Now, one could simply submit a string of all 0s in this case, which would always be compatible with any party's string, but Alice, Bob, and Chester can communicate to each other the type of their strings as well. This type will vary between the three of them, but Alice, Bob, and Chester can confirm among themselves that none of them have a sparse string, and can privately or publicly set a lower limit on the type of the string that any other agent submits, without loss of security. With this restriction it is highly unlikely that anyone other than Alice, Bob, or Chester could submit a valid password.

If a fourth party Eve attempts to eavesdrop on their measurements of qubits, then she will inevitably disrupt these measurements, introducing situations in which correlations between Alice, Bob, and Chester's qubits weaken. Alice, Bob, and Chester can account for this possibility by publically comparing a certain fraction of their final strings after running their protocol, and if these fractions of strings are not properly correlated, they can intuit that someone was eavesdropping and not use the password for anything that must be secure.

References

[1] B. Podolsky A. Einstein and N. Rosen. Can a quantum mechanical description of physical reality be considered complete? *Physical Review*, (47):777–780, 1935.

[2] Richard Cleve and David P. DiVincenzo. Schumacher's quantum data compression as a quantum computation. *Physical Review A*, 54(4):2636–2650, 1996.

[3] Oliver Johnson and Yuri Suhov. The von neumann entropy and information rate for ideal quantum gibbs ensembles. *http://arXiv.org/abs/math-ph/0109023v2*, 2002.

[4] Oliver Johnson and Yuri Suhov. The von neumann entropy and information rate for integrable quantum gibbs ensembles, 1. *Quantum Computers and Computing*, 3(1):3–24, 2002.

[5] Oliver Johnson and Yuri Suhov. The von neumann entropy and information rate for integrable quantum gibbs ensembles, 2. *Quantum Computers and Computing*, 4(1):128–143, 2003.

[6] Oliver Johnson and Yuri Suhov. The von neumann entropy and information rate for integrable quantum gibbs ensembles, 3. *Quantum Computers and Computing*, 5(1):55–64, 2005.

[7] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 2007.

[8] Sarah Croke Robin Blume-Kohout and Daniel Gottesman. Streaming universal distortion-free entanglement concentration. *http://arXiv.org/abs/quant-ph/0910.5952v1*, 2009.

[9] Benjamin Schumacher. Quantum coding. *Physical Review A*, 51(4):2738–2747, 1995.

[10] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[11] John von Neumann. Various techniques used in connection with random digits. *Applied Math Series*, (12):36–38, 1951.

[12] G. Vidal W. Dur and J. I. Cirac. Three qubits can be entangled in two inequivalent ways. *Physical Review A*, 62(6), 2000.

[13] Jacob Ziv and Abraham Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, 1977.