Quantum Darwinism and Preferred Choice of Subsystems

Mahima Kumar

Follow this and additional works at: https://scholarship.claremont.edu/pitzer_theses

Recommended Citation
Kumar, Mahima, "Quantum Darwinism and Preferred Choice of Subsystems" (2023). Pitzer Senior Theses. 187.
https://scholarship.claremont.edu/pitzer_theses/187

This Open Access Senior Thesis is brought to you for free and open access by the Pitzer Student Scholarship at Scholarship @ Claremont. It has been accepted for inclusion in Pitzer Senior Theses by an authorized administrator of Scholarship @ Claremont. For more information, please contact scholarship@claremont.edu.
Quantum Darwinism and Preferred Choice of Subsystems

A Thesis Presented

by

Mahima Kumar

To the Keck Science Department
of
Claremont McKenna, Scripps, and Pitzer Colleges
In Partial Fulfillment of
The Degree of Bachelor of Arts
Senior Thesis in Physics

May 1, 2023
ABSTRACT

Quantum Darwinism (QD) seeks to explain the emergence of the objective reality of the classical world from the underlying quantum physics. According to QD, the process of decoherence leaves a joint quantum state of a system and its environment in special states with non-generic properties. In particular, these special states allow information about the state of a system to be redundantly imprinted in many distinct fragments of the environment. The objectivity of post-decoherence states is quantified with a Partial Information Plot (PIP) exhibiting a characteristic classical plateau. The account of QD implicitly relies on a particular way of dividing the environment into sub-systems, corresponding to a choice of Tensor Product Structure (TPS) of the total Hilbert Space. Here, we investigate the role of this implicit TPS layer of structure in QD. We simulate a central system qubit independently interacting with a series of environment qubits and generate PIPs graphing post-interaction quantum mutual information between the system and environment fragments of various sizes. We then perform a unitary “scrambling” of the division into sub-environments, recalculate the quantum mutual informations, and compare the unscrambled and scrambled systems. In showing that the scrambled states no longer display redundancy, our results suggest that some choices of TPS are more optimal than others for admitting an account of Quantum Darwinism.
# TABLE OF CONTENTS

Abstract .......................................................... ii
Table of Contents .................................................. iii
Chapter I: Introduction ............................................ 1
Chapter II: Background ............................................. 4
  2.1 Density Matrices ................................................. 4
  2.2 Decoherence ..................................................... 5
  2.3 Pointer States ................................................... 7
  2.4 Quantum Darwinism .............................................. 7
  2.5 Quantum Mutual Information ................................. 7
  2.6 Partial Information Plots ...................................... 8
  2.7 Our Question .................................................... 10
  2.8 Tensor Product Structures .................................... 11
Chapter III: Worked Example ..................................... 12
  3.1 Initial State .................................................... 13
  3.2 Hamiltonian ..................................................... 13
  3.3 Time Evolution .................................................. 15
  3.4 Finding the Quantum Mutual Information .................... 16
Chapter IV: Methods ............................................... 20
  4.1 Pure vs. Maximally Mixed Initial State ....................... 20
  4.2 Implementing Scrambling ...................................... 22
Chapter V: Results and Discussion .............................. 24
Chapter VI: Conclusion and Future Directions .................. 27
Bibliography ........................................................ 29
Appendix A: Code Implementation ............................... 30
Chapter 1

INTRODUCTION

Quantum Mechanics provides a reliable calculational framework for predicting the outcomes of experiments on micro-systems, such as those at the smallest scales of atoms, molecules, and subatomic particles. Through Quantum Mechanics, we are able to understand the fundamental principles that govern the behavior of these particles and their interactions with each other and the environment. Though useful, our understanding of Quantum Mechanics is incomplete. We do not yet know how to interpret the math behind Quantum Mechanics in the limit of large systems with many degrees of freedom. In other words, the story of how the classical world we experience everyday emerges from the underlying quantum physics is unfinished.

The concepts of Decoherence and Quantum Darwinism (a refinement of Decoherence) start to help us finish the story. Decoherence describes the process by which a system loses information to its environment by snapping onto special pointer states, which then evolve predictably [1]. For an example, let us consider a baseball after it has been hit. The baseball will be our system and the photons from a nearby bright stadium light will be our environment. Every instant, many photons collide with the baseball and are deflected, with some eventually reaching the eyes of avid baseball fans.

![Figure 1.1: If the baseball begins in a pointer state (definite position state), the path of the photons colliding with it every instant is dependent on the position of the ball. Considering each photon as a “sub-environment,” the state of the photons encode information about which pointer state the ball is in.](image)

This composite system maintains coherence, as it started unentangled and remains unentangled. We can see that when a system begins in a pointer state, as shown in Figure 2.1, it will stay in a pointer state.
It will be important to note that the baseball fans (our observers in this example) will all agree on the position of the ball. We will call this agreement redundancy.

Figure 1.2: If the baseball begins in a superposition of position pointer states, the system and environment become entangled because the environment also enters a superposition.

Figure 2.2 depicts a system beginning in a superposition of pointer states. When a measurement is taken on a composite system in a superposition, it will snap onto one of the pointer states, entering the situation shown in Figure 2.1. Thereafter, the evolution of position remains predictable. The initial superposition of pointer states has “decohered.”

When a system begins in a superposition of pointer states, it will snap onto a pointer state, and thus stay in a pointer state.

Decoherence, this example, and redundancy will be revisited and discussed in more quantitative detail in Chapter 2.

Decoherence has been studied through numerous toy models presented in existing literature. In many of these models, it has been observed that post-Decoherence joint states of a system and environment are special. In these "non-generic" states, information about which pointer state the system snapped onto is redundantly encoded in fragments of the environment. This leads us to the idea of Quantum Darwinism, which aims to explain another feature of classical physics: objectivity [2] [3] [4] [5].

To reconcile the classical and quantum mechanical worlds we must tackle the idea of objectivity. Objective existence is an inherent characteristic of classical states. An unknown classical state can be measured several times without being disturbed by the measurement and all observers would agree on the outcome of the measurement. Quantum states, however, are fragile. Measurements of systems in quantum states can be demolishing, meaning they can change the state of the system, therefore
ridding it of objectivity [1]). Each observer will receive a different outcome. Quantum states are able to exist objectively only if measurements are restricted to the system’s pointer observable, which commutes with the preexisting state of the system. This will ensure that observers agree about the outcomes, and therefore a consensus about the state of the system can be reached.

Quantum Darwinism is a term coined by Wojciech Zurek and collaborators to describe how the classical world emerges from the quantum mechanical. In Zurek’s words, “Quantum Darwinism describes the proliferation, in the environment, of multiple records of selected states of a quantum system” [2].
The Program of Decoherence and its refinement, Quantum Darwinism, seek to describe how the classical world we experience everyday can be explained through quantum physics.

In Chapter 1, I briefly introduced these concepts which we will be discussing throughout this thesis. In this chapter, I will go into considerably more detail to substantiate that previous qualitative description quantitatively. I will also introduce the necessary mathematical tools of density matrices, quantum mutual information, Partial Information Plots, and Tensor Product Structures. In the next chapter, I will illustrate how these tools can be used with a concrete worked example.

2.1 Density Matrices
Density matrices are a primary mathematical tool in the formal description of decoherence. Consider a system and its environment, which begin in an unentangled state, but become entangled due to interactions. Entanglement between a system and its environment makes it so that we can no longer consider describe the system as being in a pure quantum state of its own. The system has to be expressed as a mixed state (consisting of a statistical mixture of pure states), and thus a pure quantum state vector cannot be used to describe the system.

Density matrices, and reduced density matrices, are then used to represent the measurement statistics of the system. They allow us to represent the statistics of all of the outcomes of any possible measurement performed.

We define a density matrix as a matrix representation of the density operator, such that the two terms are used interchangeably [1]. Given a pure state $|\psi\rangle$, the density operator is represented as

$$\hat{\rho} = |\psi\rangle \langle \psi|.$$  \hspace{1cm} (2.1)

Often, it is impossible to perform measurements that capture the full complexity of large or inaccessible environments. If the total Hilbert space is a tensor product of the Hilbert spaces of the system and environment, one often only can make a measurement on the system alone, due to the size and inaccessibility of the many
degrees of freedom of the environment. If we only have access to the environment, we can construct a reduced density matrix that represents only the system, by taking a partial trace over the full density matrix of the environment [1]:

\[ \hat{\rho}_S = \text{tr}_E (\hat{\rho}) , \]  

(2.2)

where

\[ \text{tr}_E = \sum_{|\psi_q\rangle \in H_E} \langle \psi_q | (\hat{\rho}) | \psi_q \rangle \]

(2.3)

and \(|\psi_q\rangle \) is the basis of \( H_E \).

This resultant reduced density matrix encodes the measurement statistics of the system alone, and therefore accounts for any influences the environment can have on the system. This tool will prove extremely useful for the topics discussed in this thesis.

### 2.2 Decoherence

Decoherence is the process by which a system entangles with its environment, specifically how a system loses information to the environment [1]. It is best to ground this explanation of Decoherence in a tangible example. Consider again a baseball, our system, and the photons colliding with it, our environment. The photons, of which there are an enormous amount, collide with the ball every instant and are deflected at a certain angle that is dependent on the position of the baseball and the initial path they take.

![Figure 2.1: The baseball begins in a pointer state of definite position, and as the photons collide with it every instant, the state of the photons encode information about which pointer state the ball is in. As the photons are measured by observers, the information becomes redundantly recorded. The baseball remains coherent.](image)

As the baseball’s position changes, the photons’ path of deflection also changes. We call the position of the ball an observable of the system, so the ball is in an eigenstate of the position operator.
Suppose the baseball can begin in one of two distinct position eigenstates, $|\psi_1\rangle$ and $|\psi_2\rangle$, described in Figure 2.1 by the purple arrow.

The initial state of the photons, $|E_0\rangle$, is what we call the ready state, before any interactions have occurred and the final state of the photons we will call $|E_1\rangle$ and $|E_2\rangle$, respectively. The possible interactions between the system and its environment are written as

$$|\psi_1\rangle \otimes |E_0\rangle \rightarrow |\psi_1\rangle \otimes |E_1\rangle \quad (2.4)$$

or

$$|\psi_2\rangle \otimes |E_0\rangle \rightarrow |\psi_2\rangle \otimes |E_2\rangle \quad (2.5)$$

We can see that the resulting $E$ state varies depending on $\psi$ [6] [7]. By measuring the path of the photons, we are able to know about the position of the book. It is important to note that this can only happen when the measurements taken are non-demolition measurements, meaning that they do not disturb the state of the system.

Now, suppose the book begins in a superposition of position eigenstates. In this case, the system and environment become entangled, because the environment will also enter a superposition, as represented in Figure 2.2 with blue and red.

![Figure 2.2: When the baseball starts out in a superposition, the superposition spreads to the environment, so entanglement occurs.](image)

However, when an observer measures the environment, the system snaps onto one of the eigenstates, and enters the same situation described previously, of a system starting out in a definite state. Thus, the mitral superposition of pointer states has "decohered" [6] [7].

When non-demolition measurements are taken of this composite system, we will repeatedly receive the same outcomes. If the baseball is in state $\psi_1$, then all
outcomes of the measurement will show us that the photons are in state $E_\infty$. The baseball evolves predictably, and there is no additional entanglement generated after interactions with the photons.

2.3 Pointer States
This only applies to what is labeled a “pointer state.” A term coined by Zurek, pointer states refer to the quantum states that are less affected by decoherence than other states, and are the quantum representations of the classical states of the system after interaction with the environment has resulted in decoherence. They are often referred to as quasiclassical states. Pointer states are the preferred states such that the loss of information to the environment is minimized [2].

Pointer observables are uniquely able to transmit information robustly about the system to the environment. The environment is then acting as a measurement device, in that information about the system is being encoded into the environment without any kind of entanglement being generated. Thus, pointer states are resistant to entanglement with the environment, and therefore resistant to decoherence [7].

2.4 Quantum Darwinism
Quantum Darwinism seeks to address the question of why pointer observables are able to realize objective existence. It argues that the objective existence of a quantum system arises as a consequence of the interactions between the system and its environment [8]. Objectivity comes about because of redundancy, the idea that the same information can be obtained by independent observers from many fragments of the environment. Therefore, we say the pointer observable is "einselected." The only states capable of producing redundancy are the einselected pointer states [4].

In order to study Quantum Darwinism, we look at mutual information to tell us how the information is transmitted from a system to its environment, when redundancy can be reached, and how we can learn all possible information about a system from its environment.

2.5 Quantum Mutual Information
Next, we need to account for the complexity of most environments, and the fact that they may not be monolithic. To do so, we model the environments as consisting of subsystems that we break into independently accessible fragments, which we denote $\mathcal{F}$. This allows us to look at the relationships between each fragment $\mathcal{F}$ and the system $\mathcal{S}$ individually. The tool we use to measure information about a system
that can be learned from $\mathcal{F}$ is called mutual information. Mutual information is, in the simplest of terms, how much each fragment $\mathcal{F}$ knows about $S$ [2]. Formally defined, mutual information is the difference between the entropies of two systems.

**Von Neumann Entropy**

The entropy we use when calculating mutual information is called the Von Neumann entropy. Von Neumann entropy is a commonly used measure for quantifying the purity of a density matrix of a system. It can be interpreted as the quantum mechanical analog of the classical statistical mechanics description of entropy.

$$S_{VN,S} = -\text{tr} (\hat{\rho}_S \log \hat{\rho}_S). \tag{2.6}$$

Using this formula, we are able to define mutual information [2]:

$$I(S : \mathcal{F}) = S_S + S_F - S_{S,F}. \tag{2.7}$$

Quantum mutual information, while just one measure of information, is rather simple to compute. Therefore, it is the method we will use to quantitatively display how much information we can learn about a system from its environment.

**2.6 Partial Information Plots**

To visualize mutual information, we create Partial Information Plots which display the mutual information between a system and fragments of its environment with respect to the size of the fragment. To create a PIP, we consider a system that has started off in an unentangled state, and let some time pass as reactions between the system and its environment occur. The time evolution of the system is governed by a chosen Hamiltonian. The PIP then evaluates the system post-interaction.

There are some key features of PIPs. If we look at the three terms in the equation to calculate mutual information, $S_S$, $S_F$, and $S_{S,F}$, we can evaluate their sum easily when none of the environment has been measured yet. $S_F$ will be 0, because no fragments have been measured, and thus $S_F$ will equal $S_{S,F}$, because the environment has not contributed any entropy yet. Therefore, the initial mutual information for a system will always be 0.

In Figure 2.3, we see an initial spike when a small fraction of the environment is measured. This indicates that the first fragments of the environment that we measure are able to give us the most information about the system.
Figure 2.3: An example of a standard Partial Information Plot of quantum mutual information vs. the fraction of the environment being measured produced by Zurek and collaborators [2]. The red plot is an example of a PIP for a composite system created by Decoherence. The green plot is an example of a PIP for a random, generic state.

Subsequently, due to redundancy, at a certain point the fragments agree about the state of the system and do not lend us any new information. Picture again the baseball with photons hitting it at every moment and being deflected. Imagine each photon hit a different observer in the stands of the baseball stadium. The first few observers to receive photons can tell us a lot of information of the position of the ball. However, every succeeding observer will tell us the same information, and have nothing to convey that a previous observer has not already told us. This is redundancy, and this is what creates the characteristic plateau in the Partial Information Plot. An key observation is that the system plateaus at $S_S$, which is all of the information about $S$ available from either $E$ or $S$. This is the limit of classically accessible information.

The last spike can only be obtained through global measurements on $S$ and nearly all of $E$. Looking at the three terms to calculate mutual information, we notice that the entropy of the system in this final state has spread to the environment such that $S_S = S_{\mathcal{F}}$, because mathematically, the pure state of a bipartite quantum system has the property that $S_S = S_{\mathcal{E}}$. The system and environment form a joint pure state with an entropy of $S_{S,\mathcal{F}} = 0$. Therefore, the mutual information spikes to $2S_S$ when all
of the environment has been measured [2].

From this we can see how a large redundancy implies objectivity. Here lies the importance of Quantum Darwinism: it accounts for the emergence of objective existence.

2.7 Our Question

Now that we have the background, let us address the research question guiding this thesis: If we “scramble” the environment by changing how the environment is divided into subsystems, does the shape of the Partial Information Plot change?

To elaborate, by "scrambling," we are referring to changing the basis of the Hilbert space in order to change the definition of our fragments of the environment. Figure 2.4 displays precisely what we mean by fragments of the environment.

Figure 2.4: An example model of a central system coupled with the surrounding environment. Each "E" denotes a subenvironment. Different groupings of subenvironments are called fragments. After scrambling, the subenvironments stay the same size, but the way they are divided up has been changed. The demarcation between system and environment remains unchanged.

In order to perform this scrambling of the environment, we will be scrambling the Hamiltonian by performing a unitary transformation on it. This will relabel the directions in the Hilbert space and allow us to manipulate the division of the environment into fragments. From here, we will produce Partial Information Plots to analyze if and how scrambling the environment affects the transmission of information between a system and its environment.

To illustrate this, we work through a concrete example in quantitative detail, and generate a corresponding PIP.
2.8 Tensor Product Structures

We must also define a Tensor Product Structure, or TPS. The tensor product structure is a concept in Quantum Mechanics that allows us to describe composite systems and their properties. A tensor product structure is defined as an equivalence class of unitary scramblings:

\[ U(2^n) \mod U_1 \otimes U_2 \otimes U_3 \otimes \cdots \otimes U_n. \]

(2.8)

It serves to formalize the division of a composite system into its subparts. In other words, the tensor product of two quantum systems represents their joint system, where the state of the composite system is described by a tensor product of the states of the individual systems [7]. For example, when we are given the individual components of a Hilbert space \( H_1 \) and \( H_2 \), we are able to tensor these parts to find the whole [9] [10]:

\[ H = H_1 \otimes H_2. \]

(2.9)

The TPS formalizes the reverse. It helps us find different factorizations of some total Hilbert space \( H \). These factorizations are not unique.

A local unitary operation will change basis within the subsystems but will not alter the division between the. In Chapter 5, we will see that local unitaries do not alter the shape of a PIP.

Previous work also explores the role of TPSes from other angles. Zanardi argues that while the choice of TPS is indeed a necessary piece of structure, it is always dictated by the particular experimental conditions at hand [ZANARDI]. Cotler argues that given a Hamiltonian alone, there is usually at most one TPS in which the Hamiltonian takes a local form, so that, it is not necessary to assume a particular TPS [11]. As such, in the existing literature, the role of TPS is a true point of controversy.
Firstly, we must establish what model we will be using. For simplicity, we will consider a system made up of qubits, which can exist in one of two states or a superposition of the two states. Our model will designate one qubit to be the system, and the rest to make up the environment. Each environment qubit will be considered a subenvironment, and will have an independent, but identical interaction with the central system qubit. Figure 3.1 shows an example of what such a system could look like.

We will begin by considering a rather simple system consisting of three qubits: one designated as the system, and two designated as subenvironments $E_1$ and $E_2$ respectively.

![Diagram](image)

Figure 3.1: An example of an n-qubit system, where $n = 5$. There is one central system qubit and four environment qubits (each is a subenvironment) that are each governed by an independent, but identical Hamiltonian. The dots represent the initial state of the system, the ground state. Each subenvironment begins in the $|0\rangle$ state, and the system begins in a superposition of the $|0\rangle$ and $|1\rangle$ states.
3.1 Initial State

We define the initial state of the system to be a superposition of two pointer states, and the initial state of the environment as the ready state, before any interactions occur. We will assume the ready state is the ground state of the environment. The ground state of a system is usually in what we call the "zero temperature limit," which is when the probability that the system is in any state other than its lowest energy state is 0. For our purposes, we assume that the environment is in the ground state of the zero temperature limit for simplicity. In the zero temperature limit, we know that the self-Hamiltonians of the environment are in the ground state. However, we can assume that the self-Hamiltonians are small compared to the interaction Hamiltonians, and are therefore negligible. This is due to the Quantum Measurement Limit.

This is also a safe assumption because this is often the most physically relevant in realistic scenarios.

For this problem, we will use an initial state that is a pure state:

\[
\psi_0 = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \otimes |0\rangle_{E_1} \otimes |0\rangle_{E_2}.
\] (3.1)

To find the density matrix of this initial state, we compute

\[
\rho_0 = |\psi_0\rangle \langle \psi_0| = \begin{bmatrix}
\frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\] (3.2)

3.2 Hamiltonian

Next, we need to choose a Hamiltonian to govern the time evolution of the system. For this particular example, we are going to use a Hamiltonian that Zurek calls \( H_{c-NOT} \) due to its role as the logic gate of a quantum circuit [12]. This Hamiltonian
takes the form:

\[
H_{c\rightarrow \text{NOT}} = \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & \frac{1}{2} & -\frac{1}{2} & 0 \\
0 & -\frac{1}{2} & \frac{1}{2} & 0
\end{bmatrix} \otimes I_{dxd}
\]  

(3.3)

where \( d \) denotes the necessary dimensions of the identity matrix to satisfy the total number of environment qubits in the system \( 2^{n-1} \). For our worked example, we will use a \( 2 \times 2 \) identity matrix because we want this Hamiltonian to induce an interaction just between the system qubit \( S \) and the environment qubit \( E_1 \). This will produce the first part of our total Hamiltonian for the system:

\[
H_{S,E_1} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}.
\]  

(3.4)

However, we have to account for interactions between the system qubit \( S \) and the environment qubit \( E_2 \). In order to do this, we will need to have another component to our Hamiltonian, and then add the two components together to create a Hamiltonian that governs the time evolution of the entire system.

We can create the second component of our Hamiltonian by acting a permutation on the first. This will effectively relabel the two environment qubits, so that we can have a Hamiltonian that describes the interaction between \( S \) and \( E_2 \).

First we compute the appropriate permutation matrix:

\[
\Pi_{1\leftrightarrow 2} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}.
\]
Applying this permutation matrix to our $H\neg\neg NOT$, we get

$$H_{S,E_2} = \Pi_{1\leftrightarrow 2} \cdot H_{c\neg NOT} \cdot \Pi_{1\leftrightarrow 2}^\dagger$$  \hspace{1cm} (3.5)

but as per the definition of a permutation matrix, $\Pi_{1\leftrightarrow 2} = \Pi_{1\leftrightarrow 2}^\dagger$.

Therefore, Equation (3.5) yields

$$H_{S,E_2} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 \\
0 & 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 \\
0 & 0 & -\frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 \\
0 & 0 & 0 & -\frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}.$$  \hspace{1cm} (3.6)

Now that we have the Hamiltonian that governs the interaction between $S$ and $E_2$, we can add it to the Hamiltonian that governs the interaction between $S$ and $E_1$. Hamiltonians are additive, so we add $tH_{S,E_1}$ and $H_{S,E_2}$ to get:

$$H_{total} = H_{S,E_1} + H_{S,E_2}$$  \hspace{1cm} (3.7)

$$H_{total} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 \\
0 & 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 \\
0 & 0 & -\frac{1}{2} & 0 & -\frac{1}{2} & 0 & 1 & 0 \\
0 & 0 & 0 & -\frac{1}{2} & 0 & -\frac{1}{2} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}.$$  \hspace{1cm} (3.8)

This is the Hamiltonian that we can now use to govern the time evolution of our entire system.

### 3.3 Time Evolution

The next step is to apply our Hamiltonian to our initial state so that we can obtain our final state. To do this, we choose a time over which we will let the system evolve that we call the characteristic time, $t_{char}$. This is given by

$$t_{char} = \frac{1}{{||H||}_2}.$$  \hspace{1cm} (3.9)
For the purpose of simplicity, we will choose our characteristic time to be $\pi$.

Now applying the Hamiltonian to our initial state, we use

$$\rho_f = U \cdot \rho_0 \cdot U^\dagger,$$

where

$$U = e^{-i \cdot H_{\text{char}} \cdot t \cdot H_{\text{total}}}.$$ (3.10)

Our problem yields

$$U = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{3} - \frac{i}{3} & 0 & \frac{1}{3} + \frac{2i}{3} & 0 & \frac{1}{3} - \frac{i}{3} & 0 \\
0 & 0 & 0 & \frac{1}{3} - \frac{i}{3} & 0 & \frac{1}{3} + \frac{2i}{3} & 0 & \frac{1}{3} - \frac{i}{3} \\
0 & 0 & \frac{1}{3} + \frac{2i}{3} & 0 & \frac{1}{3} - \frac{i}{3} & 0 & \frac{1}{3} - \frac{i}{3} & 0 \\
0 & 0 & 0 & \frac{1}{3} + \frac{2i}{3} & 0 & \frac{1}{3} - \frac{i}{3} & 0 & \frac{1}{3} - \frac{i}{3} \\
0 & 0 & \frac{1}{3} - \frac{i}{3} & 0 & \frac{1}{3} - \frac{i}{3} & 0 & \frac{1}{3} + \frac{2i}{3} & 0 \\
0 & 0 & 0 & \frac{1}{3} - \frac{i}{3} & 0 & \frac{1}{3} - \frac{i}{3} & 0 & \frac{1}{3} + \frac{2i}{3}
\end{bmatrix}. \quad (3.11)$$

Performing the time evolution, we get

$$\rho_f = \begin{bmatrix}
\frac{1}{2} & 0 & \frac{1}{6} - \frac{i}{3} & 0 & \frac{1}{6} + \frac{i}{6} & 0 & \frac{1}{6} + \frac{i}{6} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{6} + \frac{i}{3} & 0 & \frac{5}{18} & 0 & -\frac{1}{18} + \frac{i}{6} & 0 & -\frac{1}{18} + \frac{i}{6} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{6} - \frac{i}{6} & 0 & -\frac{1}{18} - \frac{i}{6} & 0 & \frac{1}{9} & 0 & \frac{1}{9} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{6} - \frac{i}{6} & 0 & -\frac{1}{18} - \frac{i}{6} & 0 & \frac{1}{9} & 0 & \frac{1}{9} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}. \quad (3.12)$$

3.4 Finding the Quantum Mutual Information

Now, we need to calculate the mutual information of the system in order to create our Partial Information Plot. Equation (2.7) tells us that the mutual information is a sum of purity entropies. Specifically, the purity entropies of the system $S_S$, the fragment $S_F$, and the system and the fragment $S_{S,F}$. 

as our final state of the system.
For this example, we are considering each environment qubit to be a fragment, so that when we take the mutual information after measuring one environment qubit, we will have measured half of the total environment. This also means that our plot will be very simple and consist of only three points. The first is when none of the environment has been measured, the second when half of the environment has been measured, and the third when all of the environment has been measured.

As discussed previously, we can deduce that the plot must start at the point \((0, 0)\) because when none of the environment has been measured, \(S_F = 0\) and \(S_S = S_{S,F}\), so \(I(S : F) = 0\).

We also know that for pure states, when all of the environment has been measured, the mutual information will go to \(2S_S\) because \(S_S = S_F\), and \(S_{S,F} = 0\).

Therefore, for this simple three-qubit system, we need only calculate the second plot point.

We start by finding the necessary partial density matrices. It will be easy to explain this step if we designate each qubit as a "slot," where \(S\) is slot one, \(E_1\) is slot two, and \(E_2\) is slot three. For this point, where we have measured half of the environment, we will need to take the partial trace over our \(\rho_f\) with respect to slot 1, slot 2, and slots 1 and 2. We get:

\[
\rho_S = \begin{bmatrix}
\frac{1}{2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{2}
\end{bmatrix}.
\]  
(3.14)

\[
\rho_{E_1} = \begin{bmatrix}
\frac{1}{2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{2}
\end{bmatrix}.
\]  
(3.15)

\[
\rho_{S,E_1} = \begin{bmatrix}
\frac{1}{2} & 0 \\
0 & \frac{1}{2}
\end{bmatrix}.
\]  
(3.16)

Next, we can use the partial density matrices with Equation (??) to calculate the purity entropies. This will give us an integer as a result!

\[
S_S = \frac{1}{2}
\]  
(3.17)
Finally, we can plot our mutual information. The second plot point will be:

\[ I(S : \mathcal{F}) = S_S + S_{E_1} - S_{S,E_1} = \frac{1}{2} + \frac{1}{2} - \frac{1}{2} = \frac{1}{2} \]  

(3.20)

And now that we have our three points, we can plot the mutual information for a three qubit system with two environment qubits.

Figure 3.2: A Partial Information Plot of mutual information vs. the fraction of the environment being measured for a three qubit system. This system is not large enough to display redundancy yet.

Since we are using a much smaller system than that displayed in Figure 2.3, we do not yet see the characteristic classical plateau that demonstrates redundancy. However, applying the same process to a five qubit system (again with one system qubit), our new plot, shown in Figure 3.3, looks quite a bit different.

Here, we see that even a small increase in the size of the system helps to demonstrate redundancy. As we increase the size of the system more, the plot will start to take the shape of the curve produced by Zurek.
We then find it useful to define a quantity $\delta$, as the tolerance, which we will use to quantify, and thereafter compare, the classicality of the plots we produce. In order to do this, we decide a value for $\delta$ that we will keep constant. Here, and for the remainder of this thesis, I will use

$$\delta = 0.2. \tag{3.21}$$

We use this by computing

$$(1 - \delta) S_S. \tag{3.22}$$

This will tell us what 80% of $S_S$ is, which we will call the threshold. Then, we find the minimum fraction of the environment measured that will disseminate 80% of $S_S$ information. Since pure states plateau at $S_S$, we can compare this number between plots to measure how quickly the plot rises to its plateau. This can then be interpreted as a measure of the classicality of the plot.

For this example, the threshold for the five qubit system is when the quantum mutual information is at 0.554518. To "score" classicality, we find where the threshold intersects with our PIP. Here, our classicality score is at 0.25, meaning that we reach the threshold when a quarter of the environment has been measured.
Chapter 4

METHODS

Figure 4.1: A crude visualization of how scrambling would redraw the lines between the subenvironments. The subenvironments will stay the same size, but the division between them will change. As shown, the division between system and environment is not affected.

The main question this thesis seeks to address is how the transmission of information from system to environment will change when we alter how the environment is divided into subenvironments. We will do this by performing a random unitary transformation to "scramble" the environment. We can think of this as effectively redrawing the lines that divide the environment, as shown in Figure 4.1. It is important to note we chose for this scrambling to only be performed on the environment, and so it will not affect the distinction between system and environment.

In order to do this, as discussed previously, I will be "scrambling" the Hamiltonian of our system.

4.1 Pure vs. Maximally Mixed Initial State

The ground state of a system is tied to a particular tensor product structure, and as such, when a scrambling is performed on that system, it is irreversible. Ultimately, the work of this thesis is to set the stage for performing a search over all possible scramblings to find the "most classical" ones, in which case, finding the ground state in an arbitrarily scrambled basis is computationally expensive.
In contrast, when we start a system in its maximally mixed ready state, or take the "infinite temperature limit," scrambling will not alter the state. In other words, the maximally mixed state is a scrambling invariant condition, where each possible state is equally likely. The quantum mutual information of mixed states at finite temperature limits is discussed in a 2009 review letter by Zurek, yet the effects of scrambling on such states has not yet been studied [13].

Following the methods outlined in the previous section, I created a PIP for a system in its maximally mixed state, shown in Figure 4.2. The form of the initial state for such a system is:

\[
\rho_0 = (|\psi_0\rangle + \langle \psi_0 |) \otimes \rho_E
\]  

(4.1)

where

\[
\psi_0 = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle),
\]  

(4.2)

\[
\rho_E = \frac{1}{d} \cdot I_{dxd}
\]  

(4.3)

for \(d = 2^{(n-1)}\).

![Figure 4.2: The resultant PIP for an 8 qubit system in its maximally mixed state. We see the initial spike and the classical plateau, but there is also a downward spike when most of the environment has been measured.](image)

This result is unexpected because of the downward spike in the PIP, which would mean that somehow, we lose information about the environment when most of the
environment has been measured. This is rather difficult to interpret, and further study is required to have a more comprehensive understanding of this plot. The previous literature addresses mixed states, but not the maximally mixed state, so it would be interesting to reconcile the existing research into partially mixed states with this new look into the behavior of maximally mixed states.

Due to this, I decided to focus my research on pure states. We will use the initial state described in Eq. (3.1).

4.2 Implementing Scrambling

In order to scramble the environment, we perform a unitary transformation on the Hamiltonian, which will change the basis of the Hilbert space and alter how the environment is divided into fragments.

We want our scrambling to be random, so I will choose a random unitary operator for the environment. Then, I will take the tensor product of a $2 \times 2$ identity matrix that represents the system with the random unitary to ensure that the system will not be affected by this transformation:

$$ U_{\text{total}} = I_S \otimes U_E. \quad (4.4) $$

We previously defined a tensor product structure as an equivalence class of unitary scramblings. This means that performing a local unitary transformation taking the form:

$$ U_E = U_{E1} \otimes U_{E2} \otimes \ldots \otimes U_{E(n-1)} \quad (4.5) $$

where $n$ is the number of subenvironments, will not change the tensor product structure. This is because the local unitary scramblings will be within the same equivalence class.

Further, quantum mutual information is basis independent for a system or fragment and therefore, this will not change the shape of the plot because we would only be changing the basis within each qubit not of the environment as a whole. Figure 5.2

Using a pre-existing function, we call a random matrix from a Gaussian Unitary Matrix Distribution that we then make Hermitian by exponentiating. Applying this transformation to the Hamiltonian by

$$ H' = U^\dagger \cdot H \cdot U, \quad (4.6) $$
we are left with the scrambled Hamiltonian. Lastly, we follow the same process outlined in Section 3 to produce a PIP for this scrambled environment and compute the classicality score.
RESULTS AND DISCUSSION

Figure 5.1: The resultant PIP for an 8 qubit system in an unscrambled, pure, ready state. This plot exhibits all of the expected characteristics for a pure state, with the initial spike, classical plateau, and final spike. From the plateau, we know that this system displays redundancy. The gray line at \( I(S:F) = 0.554518 \) indicates the threshold for this system.

From the resulting PIPs shown in Figs. 5.1 and 5.3, we are able to make comparisons between unscrambled and scrambled states. Firstly, we notice that scrambling due to a local unitary transformation has no effect on the generate PIP, and therefore does not affect the transmission of information from system to environment as seen in Figure 5.2. This result was, as previously discussed, expected.

We notice that an unscrambled state exhibits the classical quality of redundancy. However, the quantum mutual information of a scrambled state seems to exhibit an opposite behavior. Rather than the initial spike which tells us that the smallest fragments of the environment carry almost all of the total information of the system, the plot begins with a very shallow slope. This can be interpreted as showing that the smallest fragments of the environment tell us very little about the system. Then there is a drastic increase in the slope, which returns back to a plateau. This is the opposite of redundancy, and does not display classicality. Rather, this plot
Figure 5.2: As discussed previously, a local unitary transformation will not change the shape of a PIP. The threshold for this system is also at $I(S : \mathcal{F}) = 0.554518$.

Figure 5.3: The resultant PIP for a scrambled, pure state for an 8 qubit system. This plot does not exhibit the redundancy of an unscrambled pure state, but rather more closely resembles the PIP of a completely generic state. The threshold for this system is also at $I(S : \mathcal{F}) = 0.554518$.

closely resembles that of a completely random generic state, given by Figure 5.4. Therefore, we see that due to this scrambling, the system no longer admits Quantum Darwinism.
Figure 5.4: The PIP for an 8 qubit system in a random, generic state. The threshold for this system is also at $I(S : F) = 0.554518$.

Now, looking at our classicality scores, we see that for the unscrambled state, the PIP reaches the threshold when one seventh of the environment has been measured. This is when the fragment size is small, and the fragment only contains one subenvironment. However, for the scrambled state, the PIP crosses the threshold when four sevenths of the environment has been measured, or when the fragment size is four subenvironments. From this, we can reinforce that the scrambled state exhibits less classicality, because more than half of the environment has to be measured in order to learn most of the information available about the state of the system. For more classical systems, we should be able to learn about the system from much smaller fragments of the environment.
CONCLUSION AND FUTURE DIRECTIONS

In this thesis, I considered systems of \( n \) qubits, with one central system qubit and the rest as environment qubits. I implemented scrambling by performing a unitary transformation and generated Partial Information Plots for this system both with and without scrambling to compare how this would affect the transmission of information between system and environment.

In conclusion, Quantum Darwinism seems to require an extra piece of structure than previously assumed. The account of Quantum Darwinism implicitly relies on a particular way of dividing the environment into sub-systems, corresponding to a choice of Tensor Product Structure of the total Hilbert space. Our results suggest that some choices of TPSes are more optimal than others for admitting an account of Quantum Darwinism.

This work has been the first step towards performing a search over the space of all possible unitary scramblings to find which specific TPSes admit Quantum Darwinism. We eventually hope to answer the questions: For a given Hilbert space and Hamiltonian, does there exist a unique TPS in which Quantum Darwinism works or perhaps a discrete collection of such TPSes? Is there a distinction of “classical” Tensor Product Structures?

This thesis addressed only quantum systems that start in pure states, so another avenue for future research would be how to analyze the transmission of information for systems that begin in a maximally mixed state, within the infinite temperature limit.

We can also explore systems with stronger self-Hamiltonians, and much larger systems. The code I have written can be adapted and added on to expand with these extra parameters.

This research would further our understanding of quantum foundations, and help us to investigate the emergence of classical reality from the underlying Quantum Mechanics. It could also have implications within Quantum Computing, by studying the flow of information between qubits.
All in all, there is still quite a bit of research left to conduct in order to answer these questions, but this thesis provides a first step.


Appendix A

CODE IMPLEMENTATION

Here I will display all of the code used to produce the above results. The code is listed in order of the results, beginning with the maximally mixed state from Chapter 4 and then followed by the code used to produce the results in Chapter 5.

```plaintext
ClearAll["Global]*)
Get["/Users/mahimakumar/Desktop/thesis/BernDirac-main/BernDirac.wl"]
ClearAll[Subscript[H, SF1], H]

ClearAll::ssym: Subscript[H, SF1] is not a symbol or a valid
string pattern.

n = 8; Subscript[t, char] = \[Pi];
HSF1 = ({
    {0, 0, 0, 0},
    {0, 0, 0, 0},
    {0, 0, 1/2, -1/2},
    {0, 0, -1/2, 1/2}
})\[CircleTimes]IdentityMatrix[2^(n - 2)];

ClearAll[swapQubits, newRowNum];
newRowNum[col_Integer, Nqubits_Integer, i_Integer, j_Integer] :=
Module[{digits, tmp, newRowNum},
digits = IntegerDigits[col - 1, 2, Nqubits]; tmp = digits[[i]]; 
digits[[i]] = digits[[j]]; digits[[j]] = tmp;
newRowNum = FromDigits[digits, 2] + 1; newRowNum];
swapQubits[Nqubits_Integer, i_Integer, j_Integer] :=
Module[{dim = 2^Nqubits, col},
    SparseArray[
        Table[{newRowNum[col, Nqubits, i, j], col} -> 1, {col, 1, 
            dim}], {dim, dim}] ]

H = HSF1;
For[i = 3, i <= n, i++,
    H = H + (Normal[swapQubits[n, 2, i]] . HSF1 .
        Normal[swapQubits[n, 2, i]])]
```
\begin{verbatim}
MatrixForm[H];

U = MatrixExp[-I*Subscript[t, char]*H];
MatrixForm[U];

Subscript[\[Rho], E] = 1/2^(n - 1)*IdentityMatrix[2^(n - 1)];

Subscript[\[Rho], 0] = (Subscript[\[Psi], 0] .
  ConjugateTranspose[Subscript[\[Psi],
  \[0]]] )\[CircleTimes]Subscript[\[Rho], E];
MatrixForm[Subscript[\[Rho], 0]]; 

Subscript[\[Rho], f] =
  U . Subscript[\[Rho], 0] . ConjugateTranspose[U];
MatrixForm[Subscript[\[Rho], f]]; 

Subscript[\[Rho], S] = PartialTr[Subscript[\[Rho], f], {1}];

slots = Table[Range[2, i], {i, n}];

vnEntropy[\[Rho]_] :=
  Module[{nonzeroEigenvalues},
    nonzeroEigenvalues =
      DeleteCases[Eigenvalues[\[Rho]], 0]; -Sum[
        nonzeroEigenvalues[[i]] Log[nonzeroEigenvalues[[i]]], {i, 1,
        Length[nonzeroEigenvalues]}];

vonneumann[slot_] := Module[{},
  If[slot == Range[1, n], vnEntropy[Subscript[\[Rho], f]],
    If[slot == {}, -(Tr[Subscript[\[Rho], f]] Log[
        Tr[Subscript[\[Rho], f]])]),
      vnEntropy[PartialTr[Subscript[\[Rho], f], slot]]] ];

mutualinfo[slot_] :=
  vonneumann[{1}] + vonneumann[slot] - vonneumann[Prepend[slot, 1]]; 

mutualinfo[{2, 3}];
\end{verbatim}
mutual = mutualinfo /@ slots;

ListPlot[{Table[{i/n, mutual[[i + 1]]}, {i, 0, n - 1}],
          Joined -> True};

In[2509]:= ClearAll["Global"\*]
Get["/Users/mahimakumar/Desktop/thesis/BernDirac-main/BernDirac.wl"]
ClearAll[Subscript[H, SF1], H]

During evaluation of In[2509]:= ClearAll::ssym: Subscript[H, SF1]
is not a symbol or a valid string pattern.

In[2512]:= n = 8; Subscript[t, char] = \[Pi]; \[Delta] = 0.2;
HSF1 = ({
    {0, 0, 0, 0},
    {0, 0, 0, 0},
    {0, 0, 1/2, -1/2},
    {0, 0, -1/2, 1/2}
})\[CircleTimes]IdentityMatrix[2^(n - 2)];

In[2514]:= ClearAll[swapQubits, newRowNum];
newRowNum[col_Integer, Nqubits_Integer, i_Integer, j_Integer] :=
Module[{digits, tmp, newRowNum},
  digits = IntegerDigits[col - 1, 2, Nqubits]; tmp = digits[[i]];
  digits[[i]] = digits[[j]]; digits[[j]] = tmp;
  newRowNum = FromDigits[digits, 2] + 1; newRowNum];
swapQubits[Nqubits_Integer, i_Integer, j_Integer] :=
Module[{dim = 2^Nqubits, col},
  SparseArray[
    Table[{newRowNum[col, Nqubits, i, j], col} -> 1, {col, 1, dim}],
    {dim, dim}] ]

In[2517]:= H = HSF1;
For[i = 3, i <= n, i++,
  H = H + (Normal[swapQubits[n, 2, i]] . HSF1 .
            Normal[swapQubits[n, 2, i]])
  MatrixForm[H];
]

In[2520]:= U = MatrixExp[-I*Subscript[t, char]*H];
MatrixForm[U];
In[2522]:= Subscript[\[Psi], 0] = (1/Sqrt[2] Ket[0] +
1/Sqrt[2] Ket[1]); For[i = 1, i <= n - 1, i++,
Subscript[\[Psi], 0] = Subscript[\[Psi], 0]\[CircleTimes] Ket
[0];
Subscript[\[Rho], 0] =
Subscript[\[Psi], 0] . ConjugateTranspose[Subscript[\[Psi], 0]]; MatrixForm[Subscript[\[Rho], 0]];

In[2525]:= Subscript[\[Rho], f] =
U . Subscript[\[Rho], 0] . ConjugateTranspose[U]; MatrixForm[Subscript[\[Rho], f]];

In[2527]:= Subscript[\[Rho], S] = PartialTr[Subscript[\[Rho], f],
{1}];

In[2528]:= slots = Table[Range[2, i], {i, n}];

vnEntropy[\[Rho]_] :=
Module[{nonzeroEigenvalues},
nonzeroEigenvalues =
DeleteCases[Eigenvalues[\[Rho]], 0]; -Sum[
nonzeroEigenvalues[[i]] Log[nonzeroEigenvalues[[i]]], {i, 1,
Length[nonzeroEigenvalues]}]];

vonneumann[slot_] := Module[{},
If[slot == Range[1, n], vnEntropy[Subscript[\[Rho], f]],
If[slot == {}, -(Tr[Subscript[\[Rho], f]] Log[
Tr[Subscript[\[Rho], f]]]),
vnEntropy[PartialTr[Subscript[\[Rho], f], slot]]]]];
mutualinfo[slot_] :=
vonneumann[{1}] + vonneumann[slot] - vonneumann[Prepend[slot, 1]];
mutual = mutualinfo /@ slots;

In[2533]:= threshold = (1 - \[Delta])*
vnEntropy[Subscript[\[Rho], S]] // Chop

Out[2533]= 0.554518
In[2534]:= 
\[a\] = ListPlot[{Table[{i/n, mutual[[i + 1]]}, {i, 0, n - 1}]},
Joined -> True, Frame -> {True, True, False, False},
FrameLabel -> {"Fraction of the Environment Measured",
"Mutual Information"}, PlotStyle -> Blue,
GridLines -> {{}, {threshold}}];

In[2536]:= ClearAll["Global"**] Get["/Users/mahimakumar/Desktop/thesis/BernDirac-main/BernDirac.wl"]
ClearAll[Subscript[H, SF1], H]

During evaluation of In[2536]:= ClearAll::ssym: Subscript[H, SF1] is not a symbol or a valid string pattern.

In[2539]:= n = 8; Subscript[t, char] = \[Pi]; \[Delta] = 0.2; 
HSF1 = ({
{0, 0, 0, 0},
{0, 0, 0, 0},
{0, 0, 1/2, -1/2},
{0, 0, -1/2, 1/2}
})\[CircleTimes]IdentityMatrix[2^(n - 2)];

In[2541]:= ClearAll[swapQubits, newRowNum]; newRowNum[col_Integer, Nqubits_Integer, i_Integer, j_Integer] := 
Module[{digits, tmp, newRowNum},
digits = IntegerDigits[col - 1, 2, Nqubits]; tmp = digits[[i]];
digits[[i]] = digits[[j]]; digits[[j]] = tmp;
newRowNum = FromDigits[digits, 2] + 1; newRowNum];
swapQubits[Nqubits_Integer, i_Integer, j_Integer] := 
Module[{dim = 2^Nqubits, col},
SparseArray[
Table[{newRowNum[col, Nqubits, i, j], col} -> 1, {col, 1, 
dim}], {dim, dim} ]
]

In[2544]:= H = HSF1;
For[i = 3, i <= n, i++,
H = H + (Normal[swapQubits[n, 2, i]] . HSF1 .
Normal[swapQubits[n, 2, i]])
MatrixForm[H];

In[2547]:= Ufull = IdentityMatrix[2]; For[i = 1, i <= n - 1, i++,
Ufull = Ufull\[CircleTimes\]MatrixExp[-I*(RandomVariate[
    GaussianUnitaryMatrixDistribution[1, 2]])];
(*For[i=2,i<=n,i++,Ufull = Ufull\[CircleTimes\] randomUnitary];this 
  is \ 
  local unitary*)
MatrixForm[Ufull];

In[2549]:= Hscramble = Ufull . H . ConjugateTranspose[Ufull];

In[2550]:= U = MatrixExp[-I*Subscript[t, char]*Hscramble];
MatrixForm[U];

In[2552]:= Subscript[\[Psi], 0] = (1/Sqrt[2] Ket[0] +
    1/Sqrt[2] Ket[1]); For[i = 1, i <= n - 1, i++,
Subscript[\[Psi], 0] =
    Subscript[\[Psi], 0]\[CircleTimes\]Ket[\[Psi], 0]];(*random unit vector of 2^n-1*)
Subscript[\[Rho], 0] =
    Subscript[\[Psi], 0] . ConjugateTranspose[Subscript[\[Psi], 0]]; Subscript[scramble[\[Rho], 0] =
    Ufull . Subscript[\[Rho], 0] . ConjugateTranspose[Ufull];
MatrixForm[Subscript[\[Rho], 0]];
If[slot == Range[1, n], vonEntropy[Subscript[\[Rho], f]],
    If[slot == {}, -(Tr[Subscript[\[Rho], f]] Log[
        Tr[Subscript[\[Rho], f]]]),
    vonEntropy[PartialTr[Subscript[\[Rho], f], slot]]]]];

mutualinfo[slot_] :=
    vonneumann[\{1\}] + vonneumann[slot] - vonneumann[Prepend[slot, 
        1]]; mutual = mutualinfo /@ slots;

In[2564]:= threshold = (1 - \[Delta])*
    vnEntropy[Subscript[\[Rho], S]] // Chop

Out[2564]= 0.554518

In[2565]:= 

In[2566]:= 
ListPlot[{Table[{i/n, mutual[[i + 1]]}, {i, 0, n - 1}]},
    Joined -> True, Frame -> {True, True, False, False},
    FrameLabel -> {"Fraction of the Environment Measured",
        "Mutual Information"}, PlotStyle -> Blue,
    GridLines -> {{}, {threshold}}];

During evaluation of In[2567]:= ClearAll::ssym: Subscript[H, SF1]
    is not a symbol or a valid string pattern.

In[2570]:= n = 8; Subscript[t, char] = \[Pi]; \[Delta] = 0.2;
HSF1 = ({
    \{0, 0, 0, 0\},
    \{0, 0, 0, 0\},
    \{0, 0, 1/2, -1/2\},
    \{0, 0, -1/2, 1/2\}
}) \[TensorProduct] IdentityMatrix[2^(n - 2)];

In[2572]:= ClearAll[swapQubits, newRowNum];
newRowNum[col_Integer, Nqubits_Integer, i_Integer, j_Integer] :=
    Module[{digits, tmp, newRowNum},
        digits = IntegerDigits[col - 1, 2, Nqubits]; tmp = digits[[i]];
digits[[i]] = digits[[j]]; digits[[j]] = tmp;
newRowNum = FromDigits[digits, 2] + 1; newRowNum;
swapQubits[Nqubits_Integer, i_Integer, j_Integer] :=
Module[{dim = 2^Nqubits, col},
  SparseArray[
    Table[{newRowNum[col, Nqubits, i, j], col} -> 1, {col, 1, 
dim}], {dim, dim}]
]

In[2575]:= H = HSF1;
For[i = 3, i <= n, i++,
  H = H + (Normal[swapQubits[n, 2, i]] . HSF1 .
     Normal[swapQubits[n, 2, i]])
MatrixForm[H];

In[2578]:= H = SparseArray[H];

In[2579]:= random =
SparseArray[
  RandomVariate[GaussianUnitaryMatrixDistribution[1, 2^(n - 1)]
]
]

In[2580]:= (*2 is size of matrix for slots 2 through n*)
In[2581]:= randomUnitary = SparseArray[MatrixExp[-I*random]];
(*For[i=2,i<=n,i++,Ufull = Ufull\[CircleTimes] randomUnitary]; this is \ 
  local unitary*)
MatrixForm[Ufull];

In[2584]:= Hscramble = Ufull . H . ConjugateTranspose[Ufull];

In[2585]:= U = MatrixExp[-I*Subscript[t, char]*Hscramble];
MatrixForm[U];

In[2587]:= Subscript[\[Psi], 0] = (1/Sqrt[2] Ket[0] +
  1/Sqrt[2] Ket[1]); For[i = 1, i <= n - 1, i++,
Subscript[\[Psi], 0] =
Subscript[\[Psi], 0] \[CircleTimes]\ Ket[0]; (*random unit vector of 2^n-1*)

Subscript[\[Rho], 0] = Subscript[\[Psi], 0] ConjugateTranspose[Subscript[\[Psi], 0]];
Subscript[scramble[\Rho], 0] = Ufull . Subscript[\[Rho], 0] ConjugateTranspose[Ufull];
MatrixForm[Subscript[\[Rho], 0]];

In[2591]:=
Subscript[\[Rho], f] = U . Subscript[scramble[\Rho], 0] ConjugateTranspose[U];
MatrixForm[Subscript[\[Rho], f]];

In[2593]:= Subscript[\[Rho], S] = PartialTr[Subscript[\[Rho], f], {1}];

In[2594]:= slots = Table[Range[2, i], {i, n}];

vnEntropy[\[Rho]_] :=
Module[{nonzeroEigenvalues},
nonzeroEigenvalues = DeleteCases[Eigenvalues[\[Rho]], 0]; -Sum[
nonzeroEigenvalues[[i]] Log[nonzeroEigenvalues[[i]]], {i, 1, Length[nonzeroEigenvalues]}]];

vonneumann[slot_] := Module[{},
If[slot == Range[1, n], vnEntropy[Subscript[\[Rho], f]],
If[slot == {}, -Tr[Subscript[\[Rho], f]] Log[
Tr[Subscript[\[Rho], f]]]],
vnEntropy[PartialTr[Subscript[\[Rho], f], slot]]]];
mutualinfo[slot_] :=
vonneumann[{1}] + vonneumann[slot] - vonneumann[Prepend[slot, 1]];
mutual = mutualinfo /@ slots;

In[2599]:= threshold = (1 - \[Delta])* vnEntropy[Subscript[\[Rho], S]] // Chop
Out[2599]= 0.554518
In[2600]:= a = ListPlot[{Table[{i/n, mutual[[i + 1]]}, {i, 0, n - 1}], Joined -> True, Frame -> {True, True, False, False}, FrameLabel -> {"Fraction of the Environment Measured", "Mutual Information"}, PlotLegends -> {"PIP"}, PlotStyle -> Blue, GridLines -> {{}, {threshold}}];

In[2601]:= ClearAll["Global'*"]
Get["/Users/mahimakumar/Desktop/thesis/BernDirac-main/BernDirac.wl"]
ClearAll[Subscript[H, SF1], H]

During evaluation of In[2601]:= ClearAll::ssym: Subscript[H, SF1] is not a symbol or a valid string pattern.

In[2604]:= n = 8; Subscript[t, char] = \[Pi]; \[Delta] = 0.2; HSF1 = ({
    {0, 0, 0, 0},
    {0, 0, 0, 0},
    {0, 0, 1/2, -1/2},
    {0, 0, -1/2, 1/2}
})\[CircleTimes]IdentityMatrix[2^(n - 2)];

In[2606]:= ClearAll[swapQubits, newRowNum]; newRowNum[col_Integer, Nqubits_Integer, i_Integer, j_Integer] := Module[{digits, tmp, newRowNum}, digits = IntegerDigits[col - 1, 2, Nqubits]; tmp = digits[[i]]; digits[[i]] = digits[[j]]; digits[[j]] = tmp; newRowNum = FromDigits[digits, 2] + 1; newRowNum]; swapQubits[Nqubits_Integer, i_Integer, j_Integer] := Module[{dim = 2^Nqubits, col}, SparseArray[Table[newRowNum[col, Nqubits, i, j], col] -> 1, {col, 1, dim}, {dim, dim}] ]

In[2609]:= H = HSF1;
For[i = 3, i <= n, i++, H = H + (Normal[swapQubits[n, 2, i]] . HSF1 . Normal[swapQubits[n, 2, i]])]
MatrixForm[H];
In[2612]:= random = RandomVariate[GaussianUnitaryMatrixDistribution[1, 2^(n - 1)]];

In[2613]:= randomUnitary = MatrixExp[-I*random];

In[2614]:= Ufull = IdentityMatrix[2] // CircleTimes randomUnitary; MatrixForm[Ufull];

In[2616]:= Hscramble = Ufull . H . ConjugateTranspose[Ufull];

In[2617]:= U = MatrixExp[-I*Subscript[t, char]*Hscramble]; MatrixForm[U];

In[2619]:= randomunitvec = ResourceFunction["RandomUnitVector"][2^n];

In[2620]:= Subscript[\[Rho], f] = (randomunitvec) // CircleTimes ConjugateTranspose[randomunitvec]; MatrixForm[Subscript[\[Rho], f]];
vonneumann[{1}] + vonneumann[slot] - vonneumann[Prepend[slot, 1]];

mutual = mutualinfo @ slots;

\[\text{In}[2628]:=\] threshold = (1 - \[Delta]) * vnEntropy[Subscript[\[Rho], S]] // Chop

\[\text{Out}[2628]= 0.550673\]

\[\text{In}[2629]:=\] a = ListPlot[{Table[{i/n, mutual[[i + 1]]}, {i, 0, n - 1}], Joined -> True, Frame -> {True, True, False, False}, FrameLabel -> {"Fraction of the Environment Measured", "Mutual Information"}, PlotLegends -> {"PIP"}, PlotStyle -> Blue, GridLines -> {{}, {threshold}}];