A Graph Theoretic Window into the Large Scale Structure of the Universe

Sarah Bawabe

Advisor: Stephon Alexander
Brown University
Physics Department

Contents

1 Background

In today's society, many modern social media networks and apps utilize node-and-edge graphs to show patterns and subgroups amongst users. One of the questions still being asked by physicists today is whether there is a pattern to large-scale structure in the universe. Following a similar vein to these modern media networks, can the galaxies of the universe be translated into a graph to find patterns and subgroups too?

Looking at images and simulations of large-scale structure in the universe today, like the one shown in Figure 1, clear filament-like elements become visible, giving sense to the common title of "the cosmic web". From looking at this image alone, a clear comparison can be made between this structure and that of a typical graph.

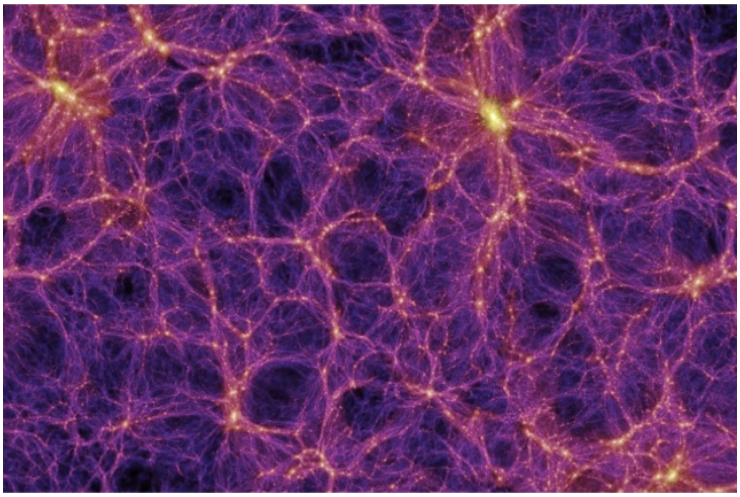

Figure 1: An image depicting the filament-like structure of the universe. This structure has a similar appearance to that of a node-and-edge graph, thus evoking a seeming correlation between the two structures. Image borrowed from [8].

1.1 Origins of Large-Scale Structure

There are three pillars of cosmology that today's modern understanding of the universe rests upon: the cosmological principle, Einstein's theory of general relativity, and Hubble's Law [3].

The cosmological principle states that the universe is homogeneous and isotropic on large distance scales. This principle has been supported by the observed homogeneity of the Cosmic Microwave Background (CMB) temperature, which suggests how the universe was once very hot and dense and thus would emit a spectrum similar to that of a typical blackbody [14]. The CMB's

temperature today is measured to be about 2.7 K, and signifies a time when the universe was once hot, dense, and highly homogeneous. This homogeneity in temperature can be seen in 2, where changes in color indicate temperature fluctuations on the order of mere $\pm$ 200 μK, or one part in 10^5.

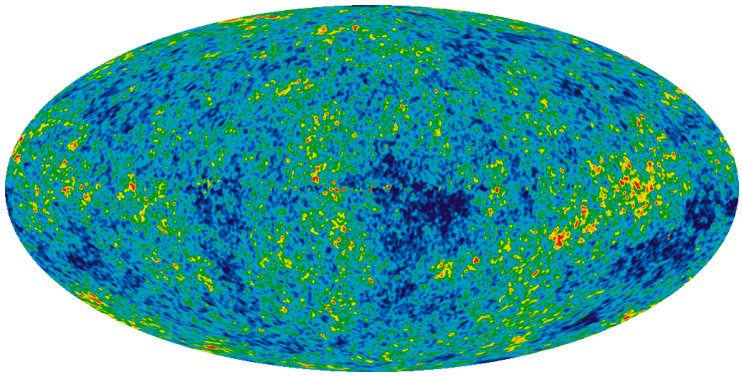

Figure 2: From the widely renowned WMAP project [15], this image shows minute temperature fluctuations (on the order of $\pm$ 200 μK, or one part in 10^5) in the CMB, which are as old as the universe itself.

Today's current understanding of the origin of the universe lies within the Big Bang theory, which theorizes that the universe began in a hot, dense state and expanded outwards (thus decreasing density) into the universe today. This theory rests upon Hubble's Law 1, which relates the expansion rate of the universe to its size through Hubble's Constant.

$$v = H_0 r \tag{1}$$

Using the CMB, the value of Hubble's Constant was measured as 68 $\pm$2 km $s^{-1}Mpc^{-1}$, while SNe measurements give a value closer to 72 km $s^{-1}Mpc^{-1}$. Though this once did not present any issues due to the wide error bars associated with these calculations, as we now enter into an era of precision cosmology, physicists have been able to calculate these values with a much smaller range of error. As cosmologists now disagree on the value of this constant and prove that error alone cannot explain their discrepancies, a problem known today as the "Hubble Tension" arose.

Since the universe is isotropic and homogeneous according to the cosmological principle, it can be determined that the universe must be described by the Friedmann-Lemaître-Robertson-Walker metric shown in equation 2, where the value of k determines whether the universe is open (-1), flat (0), or closed (+1) [9].

$$d\tau^2 = dt^2 - R^2(t)\Big[\frac{dr^2}{1-kr^2} + r^2(d\theta^2 + \sin^2\theta d\phi^2)\Big] \tag{2}$$

While the Big Bang theory predicts what the universe once looked like, further theories had to be created to solve the three main issues it arose: the

horizon problem, the flatness problem, and the magnetic monopole problem. To resolve these issues, leading theorists today turn to the theory of inflation, which hypothesizes that the universe went through a period of exponential expansion in its early stages.

The horizon problem arises due to the fact that the universe is homogeneous, but has regions which are so far apart they have no means of "contacting" one another. The Big Bang theory alone does not solve this issue, as it does not explain any points in time when these disconnected regions of the universe could have been casually connected. However, the theory of inflation resolves this issue by increasing the horizon distance, such that the early universe is fully causally connected and can be homogeneous without breaking causality [9].

The flatness problem can be solved in a similar fashion, as this inflationary era leads to a supercooling of the universe, and thus yields enough entropy to sufficiently eliminate the flatness problem [9]. For the sake of completeness, the magnetic monopole problem, which arises from the fact that cosmologists have yet to find magnetic monopoles despite their predicted existence in the Big Bang theory, is resolved by inflation since these monopoles would be spread incredibly far apart, making their discovery an incredible feat (and thus explaining why they have yet to be found).

During the CMB epoch, the energy density ($\bar{\rho}$) in the universe was mostly homogeneous and isotropic, with deviations from homogeneity on the scale of 1 part in 10^5. We refer to these deviations as density perturbations ($\delta\rho$). By introducing the theory of inflation to resolve the Big Bang's three problems, we subsequently see these density perturbations being expanded exponentially, thus yielding the small-scale inhomogeneities we see today. In our expanding and isotropic universe, the perturbations, or gravitational instabilities, seen today can be described through Equation 3 [12].

$$\ddot{\delta} - 2H\dot{\delta} - \frac{c_s^2}{a^2}\Delta\delta - 4\pi G\bar{\rho}\delta = 0 \tag{3}$$

where $\delta = \delta\rho/\bar{\rho}$. By taking the Fourier transform of Equation 3, we are able to obtain an ordinary differential equation that exists in comoving coordinate space (Equation 4).

$$\ddot{\delta_k} + 2H\dot{\delta_k} + \Big(\frac{c_s^2 k^2}{a^2} - 4\pi G\bar{\rho}\Big)\delta_k = 0 \tag{4}$$

It is useful to think of the left side of Equation 4 as corresponding to different components of the energy. The first term directly relates to the kinetic energy of the perturbation, thus dictating how quickly these perturbations are moving. The second term acts as a measure of Hubble friction, due to the fact that these perturbations are in an expanding universe which naturally wants to stretch these perturbations along with it. The third term measures the gradient energy of the perturbation, which determines its ultimate stability. This last term is critical to determining the overall behavior of the perturbations over time, and helps to derive an important quantity called Jeans length.

The Jeans length represents the critical value where gravity is able to overcome the pressure initially preventing these perturbations from growing. This value is defined in Equation 5, where k_J is defined in Equation 6.

$$\lambda_J = \frac{2\pi}{k_J} \tag{5}$$

$$k_J^2 = \left(\frac{k}{a}\right)^2 = \frac{4\pi G\bar{\rho}}{c_s^2} \tag{6}$$

Solutions to Equation 4 will be of the form shown in Equation 7, where we see that the value of ω determines behavior, where ω is defined in Equation 8. We see that for values of k above k_J, ω is real, whereas for values of k below k_J, ω is imaginary, thus directly affecting the behavior of the solution.

$$\delta \sim e^{i\omega t} \tag{7}$$

$$\omega = \sqrt{\frac{c_s^2 k^2}{a^2} - 4\pi G\bar{\rho}} \tag{8}$$

For lengths $\lambda < \lambda_J$ we observe an oscillatory behavior, but for lengths $\lambda > \lambda_J$, we see that gravity takes over and the solution becomes non-linear. Therefore, perturbation theory is no longer valid beyond this critical point and we are forced to turn to N-body simulations in order to capture the correct physics.

1.2 Data

1.2.1 Quijote Simulations

We chose to utilize the Quijote Simulations dataset [16], which contains N-body simulations for redshifts $z = 0$, 0.5, 1, 2, and 3. Since we hope to further this project to include a graph autoencoder neural network, we wanted to use a dataset with enough iterations at each redshift to sufficiently train a neural network. In this way, this project will be more easily extensible and adaptable to future continuations where the training of a network might be necessary.

For the purposes of this project, we chose to only utilize one graph from each redshift value, where the data was run on an N-body simulation in standard ΛCDM cosmology. Since this is a very widely accepted cosmological model of the universe, we chose to use it so that our data could be as closely aligned to our observed universe as possible. However, as will be discussed in Section 4, the comparison of these graphs with graphs created from datasets in different cosmologies could potentially yield interesting patterns that are undetectable when only looking at data within the same cosmological model. The cosmological parameters corresponding to our graphs are

$$\begin{array}{lll} \Omega_m = 0.3175 & \Omega_b = 0.049 & w = \text{-1} \\ h = 0.6711 & n_s = 0.9624 & \\ \sigma_8 = 0.834 & M_\nu = 0.0 \text{ eV} & \end{array}$$

as noted in [16]. We chose to utilize the halo catalogue from this simulation set.

The graphs of smaller redshifts ($z = 0$, 0.5, 1) were significantly larger than the graphs of higher redshifts ($z = 2$, 3), as can be seen in Table 1. This led to noticeable differences in computation time for graph properties, along with some more heightened statistical effects in the data, as will be seen in later plots.

Redshift (z)	# of Nodes	ℓ_{min} - ℓ_{max} [Mpc]
0	406793	6.00 - 40.00
0.5	310037	6.00 - 38.00
1	195748	6.00 - 39.00
2	44033	1.00 - 64.75
3	4776	1.00 - 79.75

Table 1: The number of nodes contained in the graphs of each redshift value, along with the ranges of linking lengths that were utilized for each redshift graph. For the larger graphs ($z = 0$, 0.5, and 1), these ranges would differ in step sizes, so that more data could be found for smaller, and thus less computationally expensive, linking lengths, while still gathering enough data at larger lengths to yield helpful results.

1.2.2 Illustris

Illustris [13, 17] is a large dataset created from a large-scale simulation of galaxy formation across different points in the universe's lifetime. This dataset is particularly applicable to the objective of this paper since it focuses on the mapping of galaxies and their overall contributions to large-scale structure. Due to time constraints, this dataset was not used within the scope of this project, but will be utilized in the continuation of this work so that our more redshift-dependent predictions and patterns can be supplemented with a wider variety of redshifts.

The Illustris dataset also provides a variety of different resolutions, where each resolution controls how finely the universe is combed for galaxies. For the beginning stages of this exploration of using graphs to find large-scale structure patterns, the lower-resolution data will be preferable due to its smaller size and focus on only larger datapoints (due to its inherent resolution). This is advantageous both in achieving faster runtimes and in searching for a coarse-grain, foundational pattern to structure. Due to the fewer datapoints, the average size of communities within the graph is expected to be smaller, since each community may be "missing" some galaxies that would have been included in a higher-resolution dataset. These differing properties within Illustris' datasets could provide interesting directions of research in future iterations of this work.

1.3 Graph Theory

An important component to this effort is understanding the basics of graph theory and how it can be utilized to find patterns in large-scale structures.

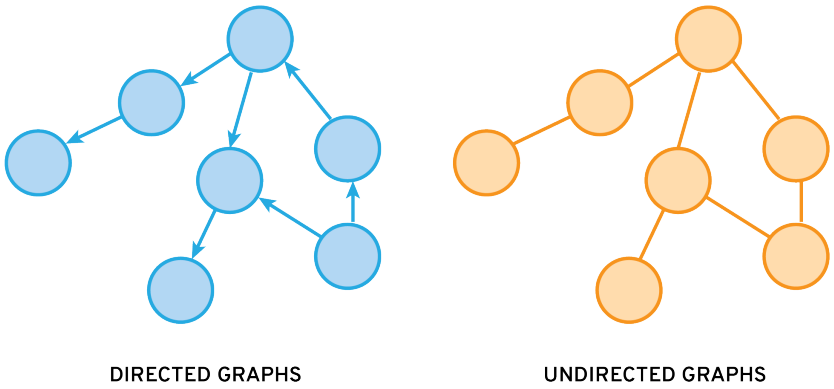

Figure 3: There are two main types of graphs: undirected and directed, where the difference lies in directionality associated with each edge. Undirected graphs do not contain directional information, whereas directed graphs do contain directional information.

A graph can be defined as a data structure that uses nodes and edges to show connections and relationships between data, where its explicit mathemetical form is shown in Equation 9.

$$\mathcal{G} \equiv (V, E);\ E \subseteq \{\{x, y\} \mid x, y \in V \text{ and } x \neq y\} \tag{9}$$

There are two main types of graphs: undirected and directed, as shown in Figure 3. Directed graphs have edges with associated directions, thus creating "Node A *leads to* Node B" relationships between nodes. However, undirected graphs do not contain directionality, and therefore create "Node A *is connected to* Node B" relationships.

For the purposes of this paper, the focus will be on undirected graphs. Not only due to the ambiguity of what directionality in the graph would correspond to physically, the main reason why undirected graphs are particularly advantageous for this paper's objective is due to the fact that they better allow for the observance of overall connectivity patterns of graphs.

1.3.1 Communities

One of the main advantages for using a graph structure is its allowance for finding communities, or substructures, in a graph. One common example of graphs being used to find these large-scale structures is in the context of social media networks, where we can think of users as nodes and their relationships (e.g. Facebook friends) to others as edges connecting them to other nodes. Looking at a graph of Hotmail users shown in Figure 4, it is discernible that amongst this large graph, smaller communities can be found. Through this search for smaller structures, research scientists at Microsoft were able to better predict

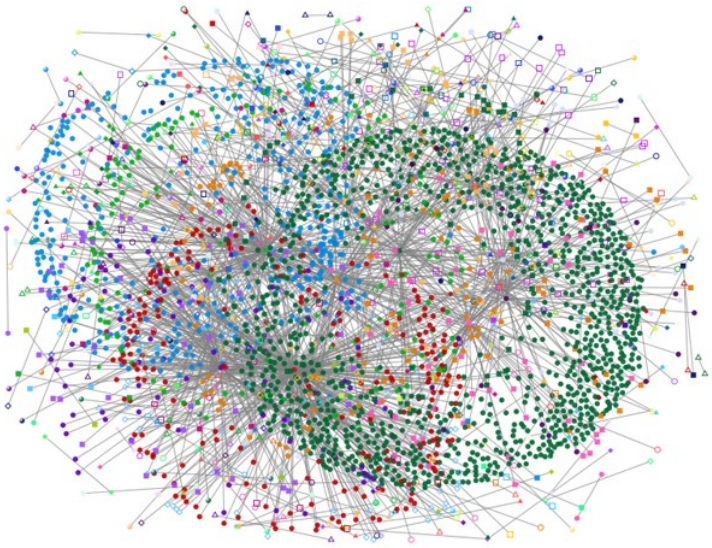

Figure 4: A graph of Hotmail's email network, which depicts the presence of its large-scale community structures. Image borrowed from [6].

which email addresses were likely spam, due to these emails' lack of connections and communities[6]. Following a similar logic, through the utilization of galaxy coordinate data to build a graph, it is hypothesized that patterns of large scale structure will appear, in the same fashion as these Hotmail connections.

A community can be defined as a group of nodes that have a higher probability of being related to each other than to other nodes in the graph [7]. While there are many algorithms in existence that can help optimize finding these communities, due to memory and time constraints caused by the extremely large size of the universe data being used, it is more practical to consider finding communities through the use of a deep learning network.

By using community algorithms to find true optimized communities on smaller graphs, these can theoretically be used as data to train a neural network, which could be taught to find these large-scale communities at a more reasonable speed given our working dataset size. Though building this neural network is beyond the time constraints of this project, we will discuss this opportunity for further exploration in Section 4.

1.3.2 Modularity

When a graph is divided into communities, an important property of the graph can be calculated: the modularity. This property is particularly interesting since it tells us essentially how "meaningfully divisible" a graph is. That is, a graph with very low modularity (≈ 0) does not contain meaningful communities that speak to patterns and structures within a graph. However, a graph with

higher modularity contains more meaningful communities that the graph can be divided into. This property can help us to determine how modular our different graphs are, which can speak to the level of small-scale inhomogeneities present in the graph.

However, while modularity and community-finding algorithms seem in theory to be a perfect way of finding structural patterns in these large graphs, they are extremely computationally expensive to compute. To emphasize, Brown's supercomputer, Oscar, was only able to calculate the modularity values for four graphs of redshift $z = 0$ in a 60-hour long job, yielding an average runtime of 15 hours/graph. Subsequently, we are forced to turn to other graph properties as well in order to complement our data and expand our search for patterns.

1.3.3 α and S_α

One of the issues that we face when creating graphs of simulation data from different redshifts is that many of their properties become redshift-dependent. In order to standardize our plots so that each property can be compared to graphs of other redshifts, we utilize a graph property called average connectivity, or α, which is defined in equation 10. This property is able to create a standard basis for the data by highlighting a property of each graph called the percolation threshold, which is the point at which a graph has as many edges as it does nodes. Since this is standard across all graphs as being the point where the largest connected sub-component of the graph (S_1) is on the order of the size of the graph, this allows us to compare our graphs on a more level field.

$$\alpha = \frac{2 \times (\text{number of edges})}{(\text{number of nodes})} \tag{10}$$

Since we will be creating edges only between nodes that are within a given radius of each other, it is not uncommon that the full graph will not be entirely connected. For this reason, we calculate the α values of the first and second largest connected components, which we designate as S_1 and S_2, respectively. By finding the properties of these largest connected components, we can gain insights into how the whole graph is structured, and as the linking length increases, this largest connected component will simply become the entire graph.

1.3.4 Clustering Coefficient ($\overline{C}$)

The clustering coefficient is a property of graphs that measures how "clustered" nodes are to each other. This value is calculated according to Equation 11, where C_i is defined in Equation 12 and k_i represents the degree, or number of neighbors, of a given vertex.

$$\overline{C} = \frac{1}{N} \sum_{i=1}^{N} C_i \tag{11}$$

$$C_i = \frac{2\Delta_i}{k_i(k_i - 1)} \tag{12}$$

This is particularly relevant in our search for large scale structure patterns as it provides a clear probe into how connected the nodes in a graph are to one another. Furthermore, this property is far less computationally expensive than community-finding and modularity algorithms, thus allowing us to gain more insights into our graphs' structural patterns both in a shorter period of time and over our full range of linking lengths.

1.3.5 Transitivity (τ_Δ)

Similar to the clustering coefficient, transitivity also speaks to the clustering properties of nodes, but instead looks at the size of the simplices contained in the graph to show more specifically how tightly connected clusters are. Transitivity is defined according to Equation 13.

$$\tau_\Delta \equiv \frac{\text{number of closed triples}}{\text{number of connected triples}} \tag{13}$$

A simplex is a way of generalizing the structure formed by a grouping of nodes, where a 1-simplex is a point, a 2-simplex is a line segment, a 3-simplex is a triangle, and so on. In order to form a simplex, every point in the cluster must be connected to every other point, resulting in a completely-triangular shape (e.g. triangle, tetrahedron, 5-cell, etc.). The relationship between simplex-number and shape formed is shown in Table 2.

Simplex #	Shape Formed
0-Simplex	Point
1-Simplex	Line
2-Simplex	Triangle
3-Simplex	Tetrahedron
4-Simplex	5-cell

Table 2: Table relating the simplex number to its corresponding shape. When calculating transitivity, these simplices and their number are found and utilized so that transitivity can give an accurate representation of how tightly clustered groups in a graph are.

1.3.6 Clique Number

Another important quantity that we will look at when analyzing our graphs is the clique number of the graph. A clique is defined as a group of vertices that forms a subgroup of a graph [18]. These substructures are particularly interesting when maximized, meaning that the clique is not a subset of any larger clique. When a graph is divided into its maximal clique components, we can find patterns within the graph's substructure which could help speak to structural patterns in the universe. Specifically, we will focus on the Clique Number of a graph, which is the size of the largest maximal clique, and the

Number of Maximal Cliques, which is the count of these maximized cliques within the graph.

1.4 Related Work

One of the most inspirational papers for our work was Hong et al. [11]. In this paper, graph analytic tools were utilized to follow a similar process to the one described in our work. However, this paper chose to compare graphs that were created within different cosmologies, or more specifically, with different Ω_m and w values. One of their most relevant plots to our work is shown in Figure 5, where plots of S_1, S_2, τ_Δ, and $\overline{C}$ as functions of linking length are shown.

This led to some interesting results and patterns, but did not speak to the formations of large-scale structures, as it did not observe data at different redshift values. For the purposes of our project, we chose to maintain the same cosmological parameters for each dataset used, but varied the redshifts. This contrast therefore will create an interesting complement to the results obtained by Hong et al., and could be extended in the future to combine aspects of both papers.

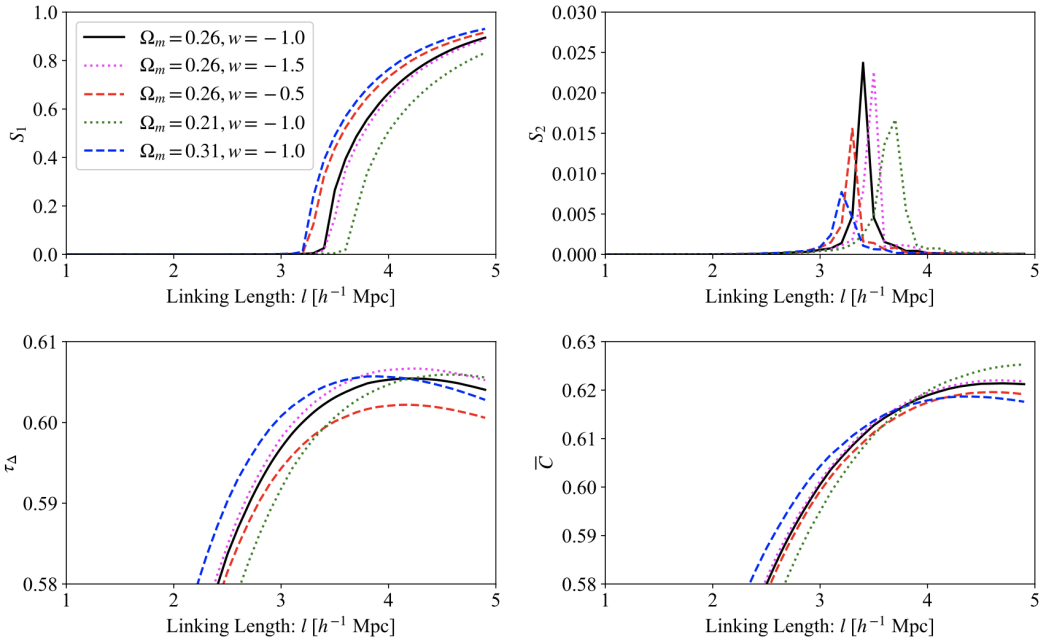

Figure 5: Plots of S_1, S_2, τ_Δ, and $\overline{C}$ as functions of linking length for graphs created in differing cosmologies. This figure is borrowed directly from Hong et al. [11], and inspired some of our choices for graph properties to look at.

2 Methods

2.1 Using Random Geometric Graphs (RGGs) as Baseline

In order to create a standardized graph for each redshift, we chose to utilize Random Geometric Graphs (RRGs) with the same number of nodes as its corresponding simulation data graph. By utilizing these RGGs we are able to create a type of "control" for all of our data, so that we would have a generalized basis to which each of our plots can be compared. These graphs are particularly applicable as a basline since they can live in any specified Euclidean space, and thus can correspond well to our simulation graph space.

RGGs are defined as a random graph that is embedded in a metric space and is constructed by assigning a random coordinate value to each vertex and choosing to connect vertices that are within a certain threshold value r of each other [4]. Since we expect the universe to be fairly homogeneous with only small-scale inhomogeneities, the utilization of randomness when creating nodes creates a similar effect within these RGGs. An example of an RGG with 500 nodes ($N = 500$) is shown in Figure 6.

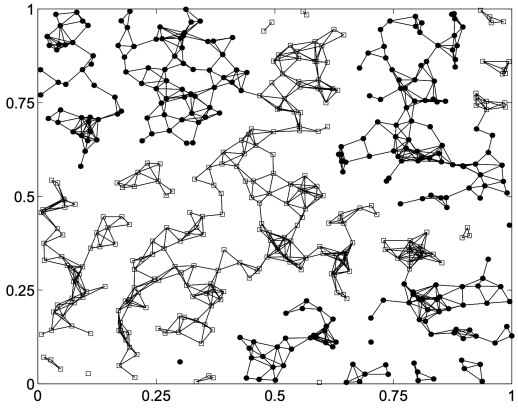

Figure 6: An example of a 2D Random Geometric Graph (RGG) with $N = 500$ nodes. The graph is shown in a unitless 1×1 square, where each vertex is plotted at its coordinate location within the square. Figure borrowed from [4].

2.2 Translating Data into Graphs

When translating the data from the Quijote Simulations into graphs [1], it's clear to define the nodes as each dark matter halo within the halo catalogue, but how we determine edges becomes more subjective.

[1] To see the code where this translation was performed, please refer to https://github.com/sarahbawabe/thesis-bawabe.

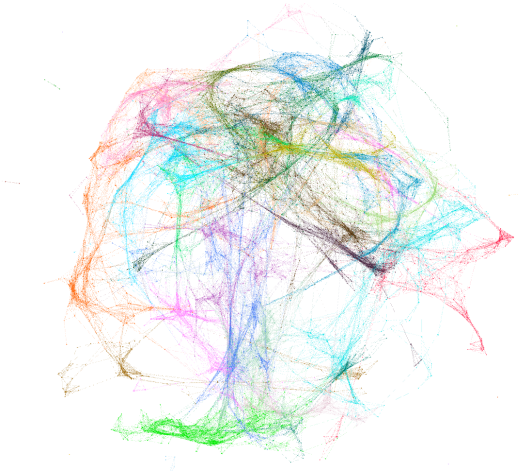

Figure 7: A piece of one of the graphs plotted using the Gephi graph visualization software. This image depicts only one-eighth of one of our full graphs, so that it was able to be handled by this visualization software. The colors of the graph represent communities within the graph.

For the sake of maintaining generality in our choice of edges, we chose to utilize a nearest neighbors radius search over a range of radii, where nodes that are within a given radius, or linking length (ℓ), of each other are connected by an edge. To optimize this process of edge creation, we used a KD-Tree nearest neighbors search algorithm, which cuts our runtime down from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$. For context, this would mean that for our redshift $z = 0$ graph, the creation of a graph at a given linking length would require $406,793^2 = 165,480,544,849$ computations if done greedily, but only requires 406,793 computations when done using this optimized KD-Tree algorithm. This is a significant and necessary choice so that our graphs are able to be created in as timely and as computationally efficient a manner as possible.

To then translate our now-determined nodes and edges into an actual graph structure, we utilized the Networkx Python package, which specializes in building graphs and performing graph algorithms [10]. This package contained code for not only building and reading the graphs we created, but also for calculating all necessary graph properties for each. [2]

To observe some of our preliminary graphs we were able to make use of the Gephi graph visualization software. Though this software was not robust enough to handle our larger graphs, we were able to visualize pieces of some of our earlier graphs to provide us with a probe of what properties might be most interesting to observe. An example of one of our graphs is shown in Figure 7.

[2] In order to streamline the process of both creating graphs and calculating each of these graph properties, we utilized Brown University's supercomputer, Oscar. All Python scripts that were used on Oscar can be found at https://github.com/sarahbawabe/oscar-scripts.

3 Results

We ended up creating a total of 1,011 graphs across all redshifts. The more precise counts are given in Table 3.

Redshift (z)	# of Graphs Created
0	113
0.5	112
1	214
2	256
3	316
TOTAL: 1,011	

Table 3: The number of graphs created for each redshift value, within the ranges of linking lengths specified in Table 1. For lower-redshifts, the creation of these graphs and subsequent computation was more computationally expensive, leading us to create fewer graphs.

3.1 S_1 and S_2

For each graph, we found the largest connected component it contained and calculated its average connectivity (α) value, as shown in Figure 8. As the linking length (ℓ) and α of the graph grew larger, these plots approached 1. While the linking length plots are shown for the sake of completeness, the α plots are more relevant to discuss, as they help to standardize our plots and support better comparisons.

However, we found a significant discrepancy between the S_1 v. α plots of the Random Geometric Graphs versus the simulation data graphs. When plotting these values for the RGGs, the curves were right on top of one other, with only slight variance, likely due to statistical randomness. However, when plotting these same values for the simulation data graphs, the curves were not on top of one other, but instead followed a distinctive pattern from left to right: $z = 3$, 0, 0.5, 2, 1. Upon further inspection, the data appears to "bounce"—beginning nearer to the RGGs' curves ($z = 3$), going further away ($z = 1$), and then returning back towards the RGGs once again ($z = 0$). This bouncing effect can be more clearly seen in Figure 9, where the colored lines are the simulation data curves, and the gray lines depict the RGG (baseline) curves.

We also looked at the second largest connected component of each graph and calculated the α value of this portion as well, shown in Figure 10. The plots of S_2 vs. α and linking length (ℓ) all eventually reach zero, as the graph becomes fully connected and no longer has a second subcomponent. To compare our simulation data to our baseline RGG data, we show an overlay of the S_2 vs. α plot on top of the RGG's S_2 vs. α data in Figure 11. These plots also depict the more precise percolation thresholds of each graph, which is the point where the S_2 achieves its maximal value.

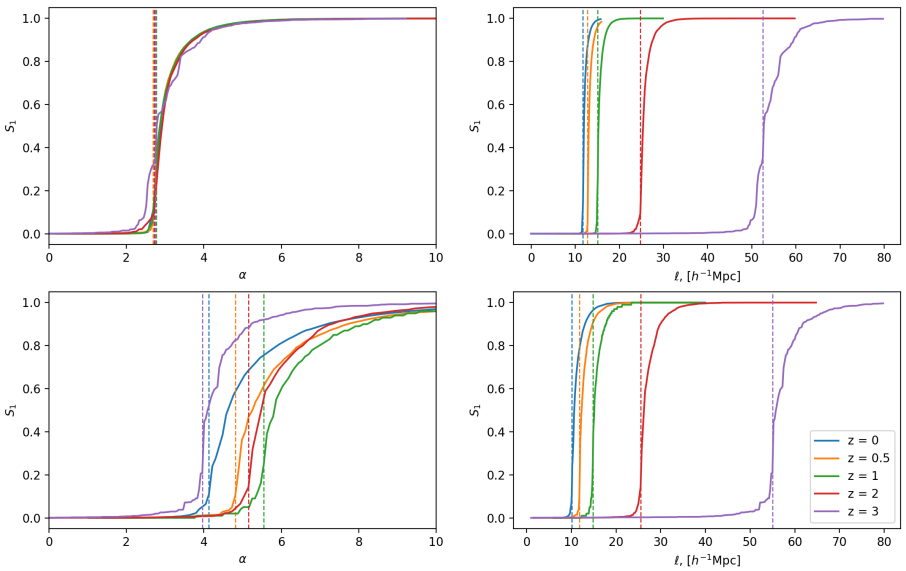

Figure 8: A plot of S_1's (the largest connected component of the graph) α value versus the α value of the entire graph for RGGs (top row) and the simulation data (bottom row). The percolation thresholds of each redshift are marked by the vertical dashed lines.

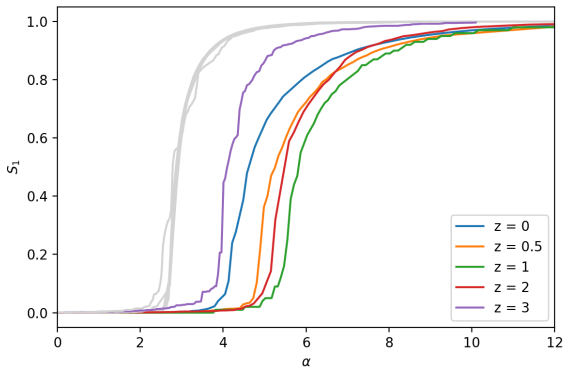

Figure 9: A plot of S_1 vs. α, where the colored lines are the simulation data curves, and the gray lines depict the RGG (baseline) curves. This figure more clearly depicts the "bouncing effect", where earlier redshifts ($z = 3$) start off closer to the baseline, then move farther away ($z = 1$), and eventually start to make their way back towards the baseline ($z = 0$).

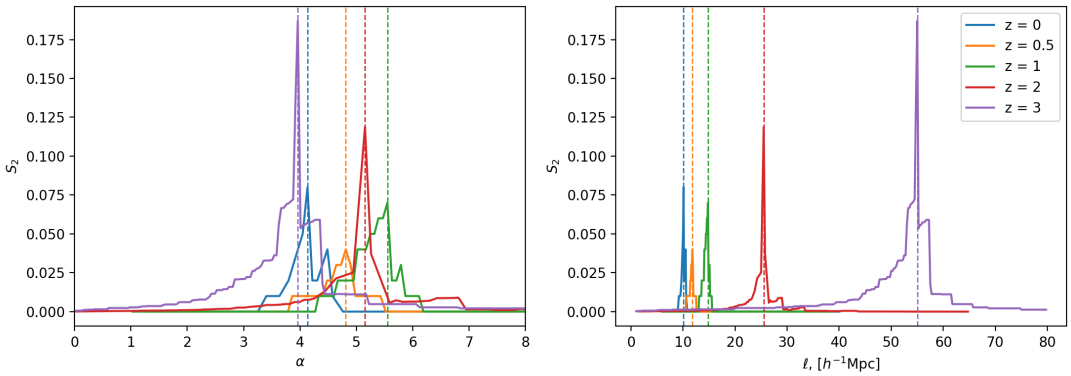

Figure 10: A plot of S_2 vs. α and S_2 vs. Linking Length (ℓ) for the simulation data graphs. The vertical lines indicate a peak for the given redshift, which is indicative of the percolation threshold of the graph.

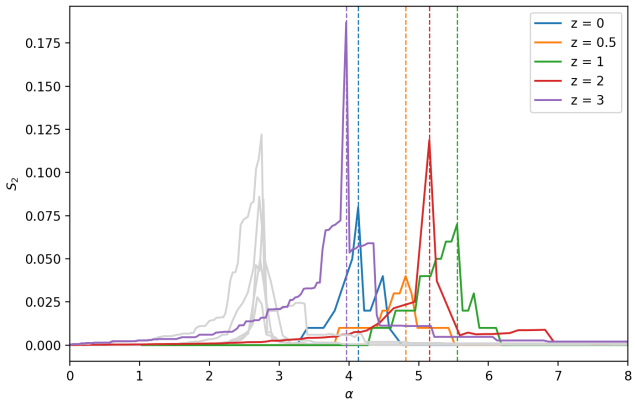

Figure 11: A plot of S_2 vs. α, where the colored lines are the simulation data curves, and the gray lines depict the RGG (baseline) curves. This figure also depicts the "bouncing effect" seen in Figures 8 and 9, where earlier redshifts ($z = 3$) start off closer to the baseline, then move farther away ($z = 1$), and eventually start to make their way back towards the baseline ($z = 0$).

3.2 Modularity

One of the most computationally expensive properties to calculate was modularity, which led our plots to be underfull for some of the smaller redshifts. For these plots, shown in Figure 12, we can observe a general behavior of modularity around the percolation threshold, where modularity begins at 1 and approaches 0 as the graph becomes more fully connected.

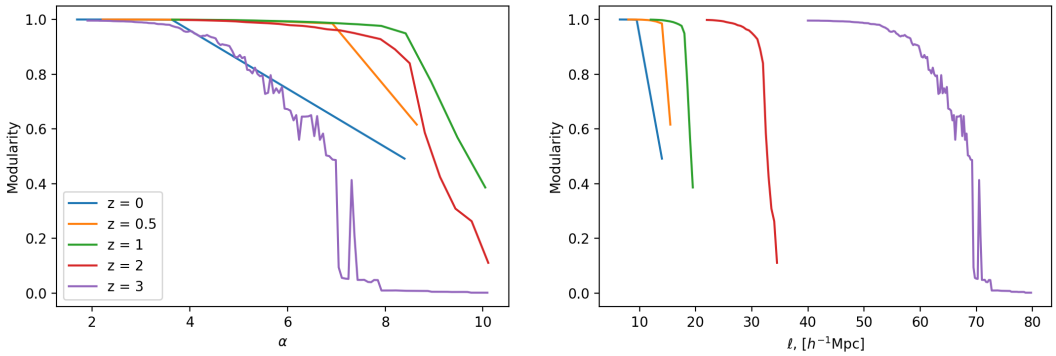

Figure 12: A plot of Modularity vs. α and Modularity vs. Linking Length (ℓ). The data for these plots was only gathered around the percolation threshold of each graph, and only for linking lengths evenly divisible by 0.5 for $z = 0$, 0.5, 1, and 2, so as to conserve and minimize runtime.

Due to the extreme runtimes of calculating modularity [3], it is important to note the number of datapoints, shown in Table 4, able to be calculated and plotted in the creation of Figure 12.

Redshift (z)	# of Modularity Datapoints
0	4
0.5	12
1	16
2	24
3	160

Table 4: The number of modularity datapoints calculated for each given redshift. These numbers were very small for redshifts $z = 0$, 0.5, 1, and 2 due to the size of those graphs, which were multiple orders of magnitude larger than the $z = 3$ graph.

3.3 Clustering Coefficient ($\overline{C}$) and Transitivity (τ_Δ)

We also calculated the clustering coefficient ($\overline{C}$), for each graph at each linking length, as shown in Figure 13. In these plots, we also see a significant difference between the curvature of the RGG plots (which seem to approach an asymptote at $\overline{C} = 0.5$) and the simulation data plots (which peak at around $\overline{C} = 0.6$ and then slope back down towards $\overline{C} = 0.5$).

[3] For redshift $z = 0$ in particular, the modularity calculation took approximately 15 hours/graph, which led to the unfortunately small number of datapoints able to be reasonably collected.

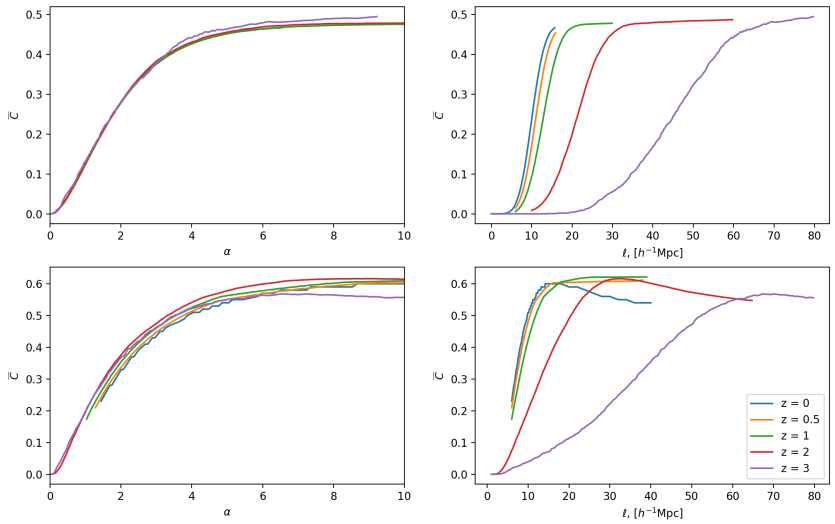

Figure 13: A plot of $\overline{C}$ vs. α and $\overline{C}$ vs. Linking Length (ℓ) for both Random Geometric Graphs (top row) and simulation data graphs (bottom row).

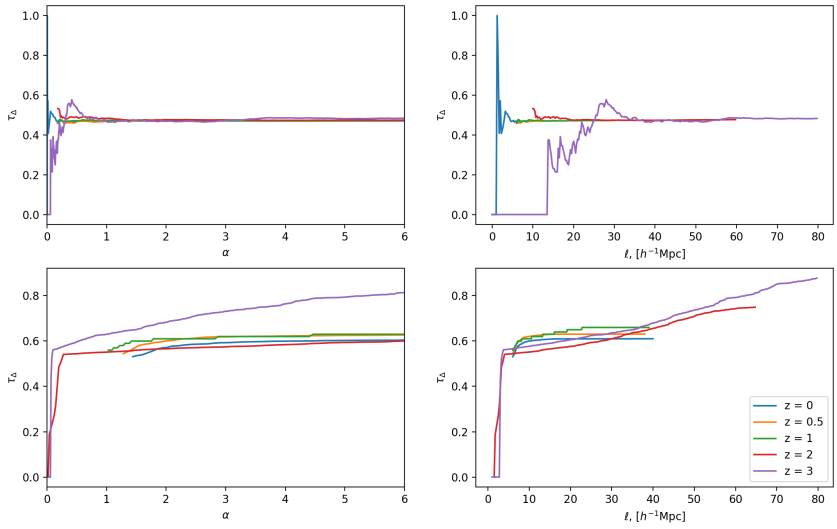

Figure 14: A plot of τ_Δ vs. α and τ_Δ vs. Linking Length (ℓ) for both Random Geometric Graphs (top row) and simulation data graphs (bottom row).

The plot depicting the transitivity of each graph at different linking lengths is shown in Figure 14. For the Random Geometric Graph, we see no early pattern for transitivity, but the transitivity soon evens out to $\tau_\Delta = 0.5$ for all

redshifts. However, we see a different picture in the simulation data graphs, as these begin to converge to $\tau_\Delta \approx 0.6$ for smaller redshifts, but $z = 2$ and 3 seem to continue increasing beyond $\tau_\Delta = 0.6$.

We also plotted the Clustering Coefficient ($\overline{C}$) versus Transitivity (τ_Δ), which is shown in Figure 15. Interestingly, this graph seemed to follow a distinctive curvature where it increased to a peak and then appeared to approach an asymptote around $\overline{C} = 0.55$.

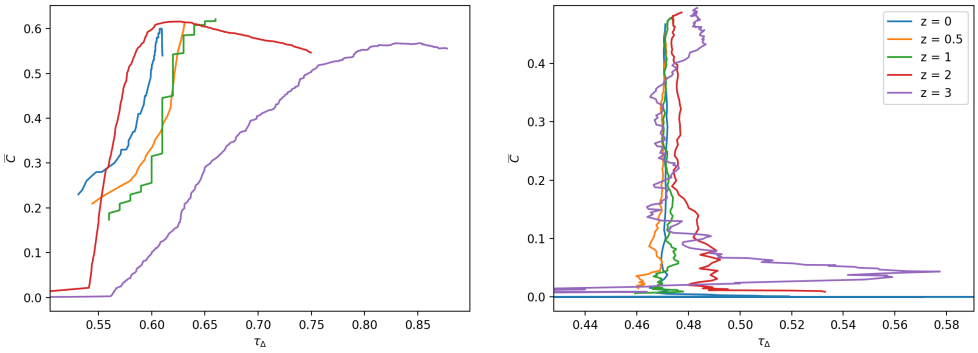

Figure 15: A plot of $\overline{C}$ vs. τ_Δ for both the simulation data (left) and the Random Geometric Graph data (right). While there is no significant pattern amongst the RGG data, we see a very distinct pattern in the simulation data plot, suggesting another difference between the large scale structure of the universe and the structure of the random graph.

3.4 Clique Number

The plot showing the Clique Number of each graph as functions of α and Linking Length (ℓ) is shown in Figure 16, and the plot depicting the Number of Maximal Cliques as a function of α and Linking Length (ℓ) is shown in Figure 17. We see in these plots that the Clique Number increases logarithmically as a function of α and nearly linearly as a function of Linking Length (ℓ). While the Clique Number plots are fairly similar in shape between the simulation data and the Random Geometric Graph, they follow very different scales, in that the RGG plot maximizes at around Clique Number = 20, whereas the simulation data plots appear to grow much quicker and peak at a higher value. We also see a slight discrepancy between the Number of Maximal Cliques plots, where the RGG plots show a slight convex-to-concave curvature compared to the only-concave curvature of the simulation data plots.

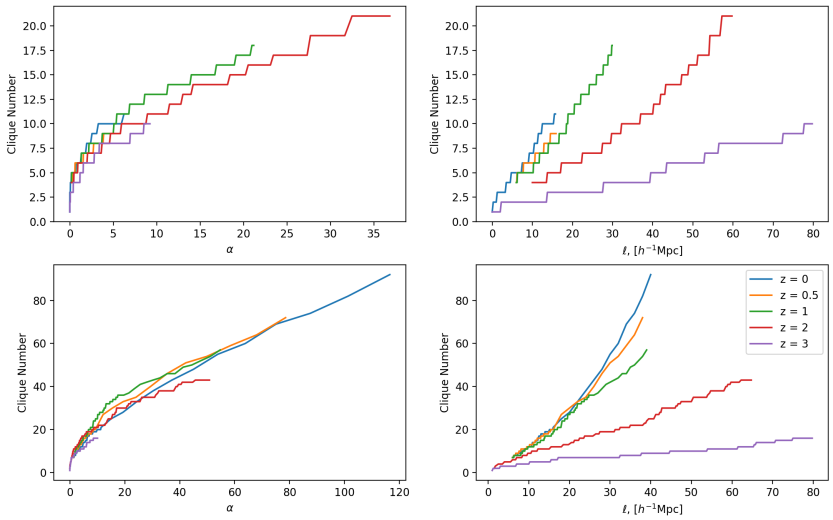

Figure 16: A plot of Clique Number vs. α and Clique Number vs. Linking Length (ℓ) for both Random Geometric Graphs (top row) and simulation data graphs (bottom row).

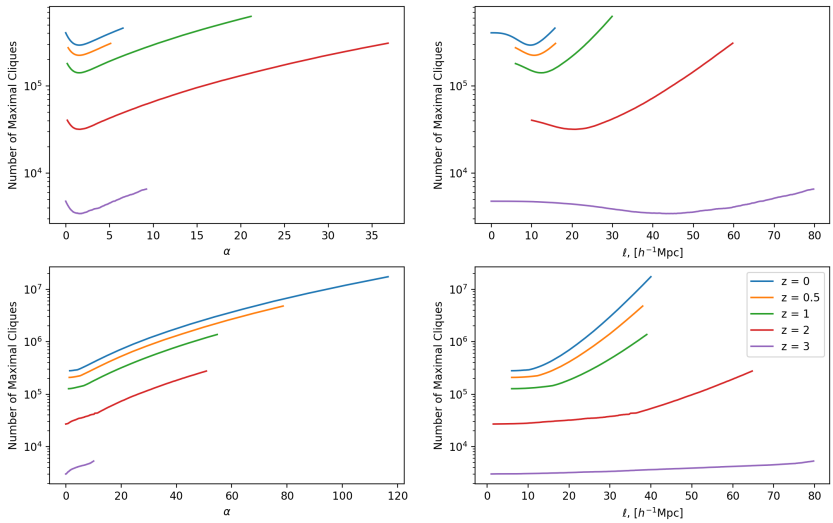

Figure 17: A plot of Number of Maximal Cliques vs. α and Number of Maximal Cliques vs. Linking Length (ℓ) for both Random Geometric Graphs (top row) and simulation data graphs (bottom row). These plots are in log-scale to allow for easier comparisons of the originally-exponential data.

4 Discussion

Through this exploration of graph theoretic applications to the search for large scale structures, we have been able to find connections between the formation of large scale structures in the universe and its corresponding graph representations.

One of the interesting peculiarities found in our plots was the order of redshift curves from left to right: $z = 3$, 0, 0.5, 2, 1, seen particularly in Figures 9 and 11. Though seemingly unordered at first, this behavior appears to be following a "bouncing" pattern, where the graphs from the earliest redshift ($z = 3$) are closest to our Random Geometric Graphs, but move progressively farther away as the redshift increases ($z = 1$), only to return back towards the RGG curves as the redshift continues to decrease and we approach modern day ($z = 0$). This bounce effect could be explained by hierarchical structure formation, which discusses how smaller structures form into larger structures over time. Through this logic, we can hypothesize that the universe began as extremely homogeneous and isotropic, leading it to appear closer in properties to a Random Geometric Graph. However, as the universe began to form smaller structures at more intermediate redshifts ($z = 1$), it began to veer farther away from random due to these more frequent small-scale inhomogeneities. Now, as the universe has continued to expand and as small structures have transformed over time into larger-scale structures, it is reasonable to postulate that the universe has begun to approach a more homogeneous state with more evenly distributed large scale structures. This would therefore lead our more modern-day universe to revert closer towards the properties of a Random Geometric Graph.

Another peculiar pattern noticed was that the redshifts $z = 2$ and 3 graphs tended to follow similar patterns with each other, and would often differ from the redshift $z = 0$, 0.5, and 1 graphs' properties. For example, why is transitivity higher for $z = 3$? How could this speak to structure formation? Though this could potentially be explained by hierarchical structure formation, where structures that begin small and close together eventually grow larger and farther apart, this question could potentially have other compelling explanations.

This exploration of graph theoretic applications has also raised many interesting questions for future works. Since the scope of this paper focuses on graphs only within the Quijote Simulation dataset, which contains only data for redshifts $z = 0$, 0.5, 1, 2, and 3, it could be insightful to explore whether these patterns continue at other, perhaps earlier, redshifts. Specifically, in order to more robustly test our "bouncing" hypothesis, the addition of more graph data within the realm of time prior to $z = 3$ could be very indicative of the truth of our explanation. It could also be insightful to search for an explanation of the bouncing hypothesis using the creation of voids over time, as opposed to large scale structures. This could perhaps be studied through the use of the void catalogues in the Quijote Simulations dataset and recreating the process followed for the halo catalogue. Through the comparison of the halo and void catalogues' graphs, it could also be compelling to search for oppositional patterns between these graphs' properties, since halos and voids are opposing in nature.

Another interesting future direction of this paper could be the investigation into using Deep Learning to discover patterns and therefore speed up some of these more computationally expensive processes. If, for example, we were to utilize a graph autoencoder to train on our graph data, we could hypothetically teach a computer to find communities in our graphs. Since community-finding algorithms are very computationally expensive and therefore extremely time-consuming on our larger graphs, by training a network to perform the same task we could in theory not only cut down on runtime and complexity, but also potentially find new insights into how communities are detected within graphs.

As was done in Alexander et al. [1], another interesting extension of training this neural network might be looking at the reconstruction loss of our simulation data graphs against the loss of corresponding different types of graphs. When an autoencoder learns to deconstruct data to its core components and then rebuild it back up (see Figure 18), we lose a certain amount of information in performing this process, called the reconstruction loss. Since we would be training this autoencoder on our simulation data set, we would expect the autoencoder to minimize loss for inputs similar to the ones it was trained on. By capitalizing on this fact, we can use reconstruction loss as a probe into testing the isomorphism between our graphs of the universe and other mathematical graphs, including but not limited to these Random Geometric Graphs. Graphs that are passed into this autoencoder and have minimal reconstruction loss could therefore be considered more isomorphic with our simulation data graphs than graphs which have larger reconstruction losses.

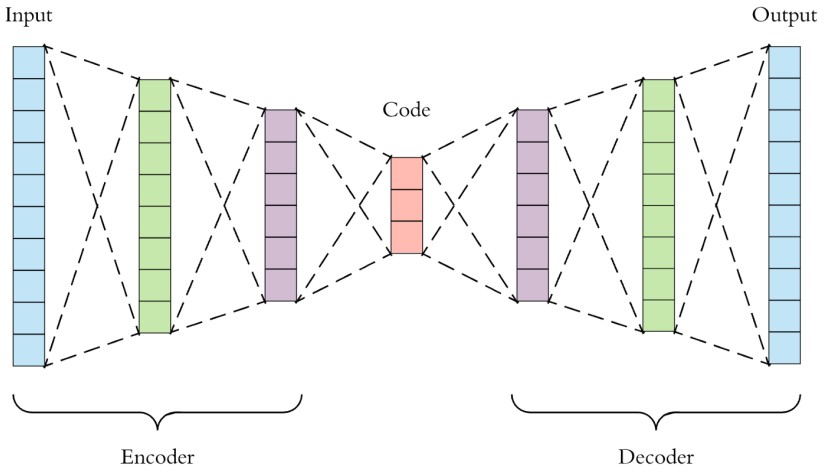

Figure 18: A model architecture of an autoencoder, where we see data being fed in, compressed to its core components, and then being reconstructed. This figure was borrowed from [5].

There are also other graph properties that were not able to be explored within the time constraints of this thesis, such as Barbour and Smolin's notion

of variety in graphs. In this paper, variety is described as a measurement for how different views of a system differ from one another. This could be calculated utilizing the adjacency matrix representation of our graphs, and applying the formulas given in Equations 14 and 15, where w_i is defined as the i'th column of the graph's adjacency matrix [2].

$$V = \sum_{i \neq j} D_{ij} \tag{14}$$

$$D_{ij} = |\vec{w_i} - \vec{w_j}| \tag{15}$$

This property could be particularly interesting since it is purely relational and non-local, therefore making it a dynamic quantity which could propose how a theory of large scale structure could work alongside the laws that govern local physics [2]. Variety can also be used as a probe for homogeneity, which could complement the work done so far.

The data also suggests that our universe is fairly similar in many properties to that of an RGG, and its divergences from RGGs can speak to the inhomogeneities in the formation of large scale structures. Since the universe is homogeneous on large scales, we would expect that the universe could be represented as a Random Geometric Graph in a coarse-grained perspective, but this statement presents another research direction entirely.

Furthermore, while we chose to utilize the Quijote Simulation data for this project due to its natural extension into machine learning, it would also be interesting to recreate these same graphs using the Illustris dataset, which is more robust and variable in its resolution, redshifts, and scales. Though using a higher-resolution dataset from Illustris sacrifices runtime due its larger data size, using this finer-grain dataset could help reveal more nuanced patterns in large-scale structure. From using this more complete dataset, we would also expect communities to be larger on average, but also more selective. When compared to its lower-resolution complement, we might be bale to better find more subtle patterns in large-scale structure that a coarser-grain dataset does not have the capacity to search for.

5 Conclusion

Though this work was largely exploratory in nature, we were able to find some insightful patterns that could inspire future works in this subject. It is theorized that a "Bouncing Effect" occurs between these graphs of different redshifts, where graphs of earlier redshifts ($z = 3$) begin closer in properties to a Random Geometric Graph, grow further away at more intermediate redshifts ($z = 1$), and return to a more homogeneous and RGG-like state as we approach modern day ($z = 0$). This effect can be explained through hierarchical structure formation in the universe, which would dictate how the structures within the universe alter its homogeneity over time. The work done in this thesis asks new and intriguing

questions and subsequently opens the door to new directions and applications of graph theory.

Acknowledgments

A special thank you to Michael Toomey for his incredible help and guidance.

References

[1] Stephon Alexander, Sergei Gleyzer, Hanna Parul, Pranath Reddy, Michael W. Toomey, Emanuele Usai, and Ryker Von Klar. Decoding dark matter substructure without supervision, 2020.

[2] Julian Barbour and Lee Smolin. Extremal variety as the foundation of a cosmological quantum theory. 3 1992.

[3] Robert H. Brandenberger. Modern cosmology and structure formation. In *Theoretical Advanced Study Institute in Elementary Particle Physics (TASI 94): CP Violation and the limits of the Standard Model*, 10 1994.

[4] Jesper Dall and Michael Christensen. Random geometric graphs. *Physical Review E*, 66(1), Jul 2002.

[5] Arden Dertat. Applied deep learning - part 3: Autoencoders. Online; accessed 8-February-2021. https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798.

[6] Brian Donahue. Researchers graph social networks to spot spammers. Online; accessed 5-February-2021. https://threatpost.com/researchers-graph-social-networks-spot-spammers-061711/75346/.

[7] Aditya Tandon et al. Community detection in networks using graph embeddings. arXiv:2009.05265v1.

[8] Evan Gough. At the largest scales, our milky way galaxy is in the middle of nowhere, date=June 8, 2017, note=Online; accessed 8-February-2021. https://www.universetoday.com/135954/largest-scales-milky-way-galaxy-middle-nowhere/.

[9] Alan H. Guth. Infiationary universe: A possible solution to the horizon and fiatness problems. https://journals.aps.org/prd/pdf/10.1103/PhysRevD.23.347.

[10] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

[11] Sungryong Hong, Donghui Jeong, Ho Seong Hwang, Juhan Kim, Sungwook E Hong, Changbom Park, Arjun Dey, Milos Milosavljevic, Karl Gebhardt, and Kyoung-Soo Lee. Constraining cosmology with big data statistics of cosmological graphs. *Monthly Notices of the Royal Astronomical Society*, 493(4):5972–5986, Feb 2020.

[12] Viatcheslav Mukhanov. *Physical Foundations of Cosmology*. Cambridge Univ. Press, Cambridge, 2005.

[13] D. Nelson, A. Pillepich, S. Genel, M. Vogelsberger, V. Springel, P. Torrey, V. Rodriguez-Gomez, D. Sijacki, G.F. Snyder, B. Griffen, and et al. The illustris simulation: Public data release. *Astronomy and Computing*, 13:12–37, Nov 2015.

[14] Barbara Ryden. *Introduction to Cosmology: Second Edition*. 2017.

[15] NASA / WMAP Science Team. Cmb images, note=Online; accessed 15-February-2021. https://wmap.gsfc.nasa.gov/media/101080/.

[16] Francisco Villaescusa-Navarro, ChangHoon Hahn, Elena Massara, Arka Banerjee, Ana Maria Delgado, Doogesh Kodi Ramanah, Tom Charnock, Elena Giusarma, Yin Li, Erwan Allys, and et al. The quijote simulations. *The Astrophysical Journal Supplement Series*, 250(1):2, Aug 2020.

[17] M. Vogelsberger, S. Genel, V. Springel, P. Torrey, D. Sijacki, D. Xu, G. Snyder, S. Bird, D. Nelson, and L. Hernquist. Properties of galaxies reproduced by a hydrodynamic simulation. *Nature*, 509(7499):177–182, May 2014.

[18] Eric W. Weisstein. Clique. From MathWorld–A Wolfram Web Resource. Online; accessed 28-March-2021. https://mathworld.wolfram.com/Clique.html.