Topological Consensus Networks

Jeffrey Morais[a,b]

[a] *Department of Physics and Astronomy, University of Victoria, Victoria, BC V8W 3P6, Canada*
[b] *BTQ Technologies, 16-104 555 Burrard Street, Vancouver BC, V7X 1M8 Canada*

E-mail: jeffrey.morais@btq.li

Abstract: Current cryptographic proofs for decentralized transactions, such as *proof-of-work* and *proof-of-stake*, face scalability and security challenges. Proof-of-work limits transaction rates to prevent inflation and multiple winners which renders larger networks inefficient, while proof-of-stake concentrates validation power in proportion to stake which enables wealthier participants to dominate governance and thus compromises network decentralization and security. We address these issues with a *modular cryptographic proof* for blockchain generation through autonomously managing closed networks. Our algorithms enable networks to partition themselves—by merging or splitting—using topologically protected classifications of network history, which naturally supports scaling. At level of consensus, networks are randomly partitioned into subsets to verify different transactions in a parallel manner, employing generalized group actions for classical networks and quantum random number generation for QKD networks with QRiNG. The combination of both yields a complete *proof-of-consensus* mechanism wherein participating *consensus networks* are dynamically checked for compliance without relying on stakes but rather on mutual benefit. The protocol's security is characterized by the evolving *trust* among network parties and external clients. We introduce a *topological formulation* to model trust dynamics in consensus networks and enable autonomous network partitioning for external multi-transaction verification. Leveraging persistent homology, we present an alternate cryptographic proof coupled with classical and quantum topological protocols for blockchain generation, offering a combinatorial method for natural scalability in transaction verification. We also briefly discuss potential hardware architectures for practical implementation.

Contents

1 Introduction

Consider a collection of parties which form a set known as a *network*, where its constituents elements are referred to as a *nodes*. Much like an ensemble of particles, these nodes can interact with one another such as participating in transactions. The network may be affected by external sources such as centralized arbiters that oversee said transactions impartially and securely. In nature we find that transactions between parties are usually arbitrated by banks which can take a cut of the transaction as a service fee. Naturally then

it would be beneficial for the nodes to minimize the degrees of freedom between themselves by removing the intermediate middle man taking the form of a centralized authority. These independent transactions, however, are subject to parties *colluding* with one another to rig transactions between nodes without central oversight. Thus, we would like a decentralized system to process transactions between nodes in a network of parties with ample security to avoid unlawful manipulation. Herein lies the concept of blockchains.

A *blockchain* is an immutable data structure whose information is stored non-locally across the network where each element of the blockchain — known as a *block* — contains the data of a transaction event. For the purposes of security, this data is encrypted via a function whose inverse is computationally infeasible to compute [known as a *hash*], and each block contains information of the previous block in the chain. The combination of the hash and information being stored in subsequent blocks prevent the transaction block from being manipulated. By construction, the blockchain is immutable: once an additional block is added to the chain, its information is fixed and cannot be altered. Now, one would wonder how to control the legitimacy of transactions when populating the blockchain if there are no central arbiters. This would have to be some form on *consensus* between the nodes of the network in which an agreement is made on the validity of the transaction and the order in which it is added to the blockchain. For methods in consensus, one could consider either open or closed networks. An *open* network is one in which external nodes can join the network freely to participate in consensus events to agree on the validity of transactions. This structure is more susceptible to colluding amongst nodes as an arbitrary amount of malicious/corrupt nodes can join to skew the consensus in their favour. This problem is attenuated in closed networks given a fixed amount of known nodes. For the more popular case of open networks, the usual consensus algorithm is known as a *proof-of-work* scheme. Given a transaction event which contains a hash [for which it is computationally infeasible to solve for its inverse], à la style of brute force, nodes compete to figure out the hash's input via sampling a random distribution of inputs. Randomly sampling hash inputs is extremely inefficient and so this process does not fall into the *complexity class* **P**, a class of problems that can be quickly solved in polynomial orders of time. Once a node randomly stumbles upon the correct input and shares it with the rest of the network, the transaction can be quickly verified by the other nodes. Thus a proof-of-work scheme for verification belongs instead to the complexity class **NP**, a class of problems of which could be verified quickly in polynomial time. Once verified and thus consensus has been made, the node which found the initial input is rewarded with a portion of the transaction, and a block is added to the blockchain. Problems present themselves when attempting to scale up proof-of-work schemes such as egregious energy costs per transaction, which are many orders of magnitude higher than one arbitrated by a bank. Furthermore, scaling induces an artificial reduction in transaction rates to prevent degeneracies in the pool of winners and prevent inflation of the underlying currency.

Could we make use of quantum computers to compute the hash inputs and rid these inefficiencies? While classically the amount of inputs used to solve the hash problem is of the order $\mathcal{O}(N)$ — where N is the size of the domain of the hash function — quantum algorithms can solve the problem more quickly given that they only need $\mathcal{O}(\sqrt{N})$ inputs

[such as with the use of Grover's algorithm]. Why not then make use of quantum computers for consensus in networks one might ask, given a potential quadratic speedup? The problem with using quantum computers for solving unstructured problems with inverting functions is that it violates *progress-free condition*. This means that nodes that have solved previous block puzzles would have an advantage over ones that have just joined to solve the current hash input. What we *can* do to make use of the advantage that quantum algorithms holds over classical computations is by sampling. Recall that the use of a quantum theory is that one can store information non-locally through entanglement for which there is associated uncertainty. This presents quite the incentive to save on computational demand as one would not need to collapse the state until measurement and so working with a state in superposition [which contains all possible information of the system] is more efficient. This of course, comes with its own share of problems with scalability as the uncertainty increases further and the need for quantum error correction and fault-tolerant codes come into play. Unlike a decision problem in which a precise solution is given [the hash input], with sampling you take measurements from a large superposition and converge to the solution given some estimated probability. One for instance could make use of *boson-sampling*, a consensus algorithm that estimates [via convergence] the expectation value of matrix permanents for boson scattering events. This could be implemented as a proof-of-work scheme for networks verifying the validity of a transaction when populating a blockchain which makes use of the quantum advantage.

Now, boson-sampling falls into its own complexity class **BosonSampP** and is thought to be strictly contained within the class of sampling problems that can be efficiently solved on a quantum computer, **SampBQP** [1]. Although these problems are sampled more efficiently with a quantum framework over that of a classical one, the fact still remains that open networks are vulnerable to external manipulation of consensus outcomes, and proof-of-work schemes are exceedingly unscalable. It is for these reasons we instead consider an alternate consensus scheme which makes use of closed networks. A *closed* network is one in which the amount of nodes [and the fraction of which are malicious] is fixed. For the purposes of consensus, we would like a scheme that is highly resistant to malicious nodes colluding to manipulate the outcome of consensus by forcing an unlawful majority. In this endeavor we prevent this by splitting these malicious nodes into subsets and taking local consensus from these sets. Such a scheme is known as a *proof-of-consensus* scheme. To the maximal degree, the allocation of these subsets must be *quantum random* as to disperse these malicious parties among honest ones and suppress their strategy. We differ to quantum randomness as classical randomness is ultimately deterministic unlike the collapse of the wavefunction is not. If a subset of the network forms consensus, it reflects the will of the majority and the colluding minority will fail. In order to ensure an honest consensus event, we test the compliance of nodes in sub-networks which reflects the amount of trust nodes have in each other. Networks which are quantum randomly assigned to sub-networks for consensus and whose compliance is tested are known as *consensus networks*. Unlike proof-of-work schemes, no transactional stake must be lost as it instead relies on mutual benefit: for every transaction a node requests to be verified, so too must they participate in an equal amount of transaction verifications. Furthermore, proof-of-consensus schemes do

not run into the scalability issues and high energy costs of proof-of-work schemes. To characterize the security of such a protocol we will need to mathematically describe the trust between nodes with a consensus network over time. Taking inspiration from the non-local aspects of quantum theory, for efficiency we will want to make use of a *global theory* [unlike a theory in which nodes share local information about transactions]. For a global theory one can employ *topology*[†] which captures the global aspects of a space [in our case the space of networks] in a manner that does not depend on local aspects such as geometry. The use of topology moreover also allows us to have a generally covariant theory [a theory which is invariant under diffeomorphisms of the space] which allows us to characterize information through topological invariants. In this paper we prescribe a topological characterization of trust [and hence security] in consensus networks used in proof-of-consensus schemes for efficient transaction verification. First we motivate the topological construction of consensus networks and how breaches in its trust create bifurcations in the form of cobordisms. We then describe how independent networks can combine with each other based on trust given by topological invariants. Then we look at the presenting this framework as a service to clients through the intersection of trust between the client and the network, and the network with itself. Finally, we come up with a protocol for autonomous networks to evaluate others and combine with a topological classification of networks [and their associated history]. This gives us a natural way to scale up networks for a more efficient service of verification without running into the inefficiencies as suffered by proof-of-work schemes.

2 Classical modular cryptography

2.1 Discrete consensus networks

Here we present the construction of discrete consensus networks with the use of metric spaces. From this we describe how to form simplicial complexes to study the topology of the network, as well as come up with algorithms that automatically partition the networks solely based on the trust of the nodes. This is done to maximize the amount of trust in each sub-network for efficient consensus protocols.

2.1.1 Trust as metric distance

One considers a network of individuals as a *discrete* set of data points typically immersed in or residing on a Euclidean manifold. Many tools available to us to study its properties — such as geometry or topology — require a continuous space from which we define functions, however it remains that individuals are not smeared over a continuous space; nodes in a network are *distinct* and *unique*. How then do we reconcile this? Indeed there exists discrete theories of geometry/topology nonetheless at the deficit of topological invariants that only exist for continuous spaces. In spite of that, one can look at the *underlying*

[†] *Differential topology* characterizes the structures of manifolds that have only trivial local *moduli* [parameter spaces that are typically quotient spaces] while *differential geometry* studies structures that have non-trivial local moduli. Points in moduli space correspond to solutions of geometric problems which are identified if isomorphic. An example moduli space is the space of flat metrics on a Calabi-Yau manifold.

continuous structure of discrete objects from which we can utilize these continuous techniques while gaining additional information along the way. To do so, we will make use of what are known as abstract simplicial complexes to study the topology of networks at different scales via what is known as persistent homology. We will see that these give us a natural prescription of trust between networks and its autonomous sub-clustering through characterizing distances [2].

Consider a set M with some notion of distance between its points given by a map $\rho : M \times M \longrightarrow \mathbb{R}_+$. Together this forms a *metric space* (M, ρ) which adheres to the following conditions [for points $x, y, z \in M$]:

i) $\rho(x, y) \geq 0$ and $\rho(x, y) = 0 \iff x = y$,

ii) $\rho(x, y) = \rho(y, x)$,

iii) $\rho(x, z) \leq \rho(x, y) + \rho(y, z)$.

This will form our *ambient space* of nodes which is endowed with a set of compact subsets $\mathcal{K}(M) \subseteq M$. We take one such subset to be our *network* of discrete nodes as $\mathcal{N} \in \mathcal{K}(M)$. Ignoring effects of curvature on the space, as well as the possibility of a temporal dimension [such as Lorentzian manifolds which we will consider later], we take the ambient space to be flat d-dimensional Euclidean space such that $M = \mathbb{R}^d$. A measure of distance between nodes in this space will correspond to the amount of **distrust**, r, they have in one another, so we say that ρ naturally encodes this [for two nodes x, y, we say the distrust of x in y is $r(x, y) = \rho(x, y)$]. It is noted that we use the terms distrust and trust interchangeably but usually refer to the former. For the case of the distrust of a node in another network or a sub-clustering of its own network, there is a subtlety that must be addressed. First we consider the distance between a node $x \in \mathcal{N}$ and some sub-clustering of its network $\mathcal{N}' \subset \mathcal{N}$, given by the map $d : M \times \mathcal{K}(M) \longrightarrow \mathbb{R}_+$ along with its associated representation:

$$d(x, \mathcal{N}') = \inf_{y \in \mathcal{N}'} \rho(x, y). \tag{2.1}$$

Here we take the *infimum* [of the distance ρ] which is the minimum of a set that allows for interpolation near its boundary should the minimum not exist in its interior. In this sense an infimum is a *maximization* of a set of minima outside the set such that it is closest to the original set's interior. For our purposes of distrust, it would not make sense to define the distrust of a node in $\mathcal{N}'$ as the minimum distance [distrust] it has with an element of $\mathcal{N}'$. This is because we can add an arbitrary amount of points infinitely far away from the reference node which would make the overall distrust reputation of $\mathcal{N}'$ diverge. Instead, we say that the distrust a node has in a sub-clustering or independent network is an average over their point-like separations [individual distrusts] and write it as the weighted sum:

$$r(x, \mathcal{N}') = \frac{1}{|\mathcal{N}'|} \sum_{y \in \mathcal{N}'} r(x, y). \tag{2.2}$$

What about a network-wide definition of distrust; how do we define the distrust between two independent networks [or rather two sub-clusterings of an overall network]? First we consider the separation between two independent networks $\mathcal{N}, \mathcal{N}' \in \mathcal{K}(M)$ through what is known as the *Hausdorff distance*. This is given by the map $d_H : \mathcal{K}(M) \times \mathcal{K}(M) \longrightarrow \mathbb{R}_+$ and its associated representation[†]:

$$d_H(\mathcal{N}, \mathcal{N}') = \max\left\{ \sup_{x \in \mathcal{N}} d(x, \mathcal{N}'), \sup_{y \in \mathcal{N}'} d(y, \mathcal{N}) \right\}. \tag{2.3}$$

If take this to encode distrust, we would say that the distrust between two networks is characterized by the lowest instance of distrust across the nodes in each network. Again this is less precise than sampling all pairs of distrust between the networks as we can add an arbitrary amount of high distrust instances between the networks and this information would get lost in the minimization. Instead we encode distrust of the network $\mathcal{N}$ in $\mathcal{N}'$ as the lower of the collective opinion it has in the other:

$$r(\mathcal{N}, \mathcal{N}') = \sup_{x \in \mathcal{N}} r(x, \mathcal{N}'). \tag{2.4}$$

The distinction here is that in order for $\mathcal{N}$ to distrust $\mathcal{N}'$, we require *all* nodes in $\mathcal{N}$ to individually distrust $\mathcal{N}'$. For a network to distrust another — meaning the above exceeds a security parameter δ in the form $r(\mathcal{N}, \mathcal{N}') > \delta$ — by our construction above this implies that $r(x, \mathcal{N}') > \delta \ \forall x \in \mathcal{N}$. Another subtlety which must be addressed is that distrust is subjective for every node and so the definition of the security parameter δ would vary for every node. Henceforth we make the replacement $\delta \longrightarrow \delta_x$ where δ_x is the security parameter vector whose components are respective to nodes $x \in \mathcal{N}$. Thus, for a network $\mathcal{N}$ to distrust another network $\mathcal{N}'$, we require the following to hold:

$$r(\mathcal{N}, \mathcal{N}') > \delta \ \text{ and } \ r(x, \mathcal{N}') > \delta_x \ \forall x \in \mathcal{N}, \tag{2.5}$$

for some global security parameter δ as agreed upon by the network $\mathcal{N}$. Note that with all distrust functions — whether between two nodes or two networks — it isn't necessarily symmetric under swapping its arguments. That is to say a node can distrust another node more than that node distrusts it [and this logic can be extended to the case of a set of networks].

Finally, we must make a notion of distrust between networks that reside in different ambient spaces. This will be useful when we compare the time evolution of different independent networks [later known as network histories] for autonomous combination, or the difference in distrust between a network of nodes and clients. Mathematically this would be computing some distance between networks that are compact subspaces of *different*

[†]At first it might not seem intuitive why one would take the maximum of a *supremum* [whereas an infimum is the greatest lower bound of a set, a supremum is its smallest upper bound] of a distance as in physics distance measures are usually minimized such as with *Hamilton's principle*. The supremum of distance separations act as radii of the covers over the networks which lets us define notions of distance. Then we take the maximum of the two such that the cover contains both networks. This is precisely the distance between the sets.

metric spaces. To do so we must discuss what are *isometries*. An isometry is a distance-preserving transformation between metric spaces that is usually bijective. Two compact metric spaces, (M, ρ) and (N, σ) are said to be isometric if there exists a bijection $\phi : M \longrightarrow N$ that preserves distances: $\rho(x, y) = \sigma(\phi(x), \phi(y))$ for any points $x, y \in M$ [and thus any corresponding points $\phi(x), \phi(y) \in N$]. The notion of distance between two independent metric spaces is precisely the measure of how far they are from being isometric. We characterize this with the use of the *Gromov-Hausdorff* distance, which is the map $d_{GH} : M \times N \longrightarrow \mathbb{R}_+$ with the associated representation:

$$d_{GH}(M, N) = \inf_{P} \inf_{\phi,\psi} d_H\Big(\phi(M), \psi(N)\Big). \tag{2.6}$$

There is quite a bit of technicality in this line so let's break it down. Here we consider isometric embeddings of metric spaces (M, N) in some ambient metric space (P, γ) via the maps $\phi : M \hookrightarrow P$ and $\psi : N \hookrightarrow P$. Moreover, the Hausdorff distance d_H is with respect to the distance measure γ of P [it is usually written as d_H^P, however we will drop this notation and infer from context]. In this sense we are minimizing the Hausdorff distance both over all possible embeddings of the metric spaces (M, N) in P, and over all possibilities of P [from this it follows that $d_{GH} \leq d_H$]. Now for the distrust between two independent networks of different ambient spaces, we instead have:

$$r(\mathcal{N}, \mathcal{M}) = \inf_{\mathcal{P}} \inf_{\phi,\psi} r\Big(\phi(\mathcal{N}), \psi(\mathcal{M})\Big), \tag{2.7}$$

where $\mathcal{P}$ is some ambient larger network and we instead have network embeddings $\phi : \mathcal{N} \hookrightarrow \mathcal{P}$ and $\psi : \mathcal{M} \hookrightarrow \mathcal{P}$ [for networks $\mathcal{N} \in \mathcal{K}(N)$, $\mathcal{M} \in \mathcal{K}(M)$, and $\mathcal{P} \in \mathcal{K}(P)$, respectively]. Now that we have complete notions of distrust on the level of nodes, networks in the same ambient space, and networks in different ambient spaces, we move on to looking at their demeanor.

How do these different grained trust functions change the structure of the network? Adhering to the discrete nature of networks we say that $\mathcal{N}$ is an *multi-index set* which labels the nodes and their underlying characteristics. Whereas before our continuous network was an element of compact subsets of the ambient space $M = \mathbb{R}^d$, instead we look at a discrete subset of the ambient space $Q \subset M$ where $Q = \mathbb{Z}^d$. This means that an element i of a network $\mathcal{N}$ would look like $i = (i_1, \ldots, i_d)$. For our case of consensus networks, we will consider the case of $d = 1$, meaning a network $\mathcal{N}$ has elements that simply label the nodes. This effectively gives us a prescription to shift from continuous to discrete via a coordinate transformation: $(x, y) \longrightarrow (i, j)$. A node i will have a subjective interpretation of distrust in every other node j in $\mathcal{N}$; we capture this information in a matrix $r(i, j) \equiv r_{ij}$, which we call the *trust matrix* of the network $\mathcal{N}$. It assumed that nodes have full trust in themselves and thus diagonals trivial vanish as $r_{ii} = 0$. Furthermore, to make use of the index conventions of discrete parameters we modify our notation for the grained trust functions:

$$r(x,\mathcal{N}') \longrightarrow r_i(\mathcal{N}') = \frac{1}{|\mathcal{N}'|}\sum_{j\in\mathcal{N}'} r_{ij}, \;\; r_{\mathcal{N}'}(i) = \frac{1}{|\mathcal{N}'|}\sum_{j\in\mathcal{N}'} r_{ji}. \tag{2.8}$$

Here we present the distrust node i has in the network $\mathcal{N}'$ and vise versa [previously written as $r(i,\mathcal{N}')$ and $r(\mathcal{N}',i)$, respectively]. Nodes can make *decisions* based on condition of this distrust, such as deciding to cooperate with a sub-network given by the boolean function $f_i : \mathcal{N}' \longrightarrow \{0,1\}$. We can represent this as a cutoff for acceptable distrust as the following:

$$f_i(\mathcal{N}') = \begin{cases} 1,\, r_i(\mathcal{N}') \leq \delta_i \\ 0,\, r_i(\mathcal{N}') > \delta_i, \end{cases} \tag{2.9}$$

where δ_i is some security parameter specified by the node i. The case $f_i = 1$ *implies* the existence of some subset $\mathcal{X} \subseteq \mathcal{N}'$ that node i wants to cooperate with. We can denote the subset as the following:

$$\mathcal{X}_i(\mathcal{N}') = \{j \in \mathcal{N}' \;:\; r_{ij} \leq \delta_i\}. \tag{2.10}$$

After a round of consensus occurs this gets updated and *from the perspective* of the i-th node there is a natural bifurcation into the subset they want to cooperate $\mathcal{X}_i$ with and its complement $\bar{\mathcal{X}}_i = \mathcal{N}' \setminus \mathcal{X}_i = \{i \in \mathcal{N}' | i \notin \mathcal{X}_i\}$ [i.e. the rest of the network that it doesn't want to cooperate with]. We present this updated network as the following bifurcation or splitting:

$$\mathcal{N}' \xrightarrow{f_i} \mathcal{X}_i \sqcup \bar{\mathcal{X}}_i. \tag{2.11}$$

In general, each of the subsets in general can be composed of an arbitrary amount of sub-components, such as $\mathcal{X}_i = \coprod_a \mathcal{X}_i^a$. This gives us a general multi-set bifurcation of the network based on the preference of node i which can be written as:

$$\mathcal{N}' \xrightarrow{f_i} \coprod_a \mathcal{X}_i^a \coprod_b \bar{\mathcal{X}}_i^b \tag{2.12}$$

Going up a scale in the graining of trust, what about bifurcations as a result of the preference of an entire network and not a single node? For this we modify our previous notation for distrust between networks of the same ambient space as:

$$r(\mathcal{N},\mathcal{N}') \longrightarrow r_{\mathcal{N}}(\mathcal{N}') = \sup_{i\in\mathcal{N}} r_i(\mathcal{N}'). \tag{2.13}$$

Thereafter, we can define a set-wise [as opposed to a point-wise] bifurcation as a generalisation of our former binary decision outcome function:

$$f_{\mathcal{N}}(\mathcal{N}') = \begin{cases} 1, \, r_{\mathcal{N}}(\mathcal{N}') \leq \delta_{\mathcal{N}} \\ 0, \, r_{\mathcal{N}}(\mathcal{N}') > \delta_{\mathcal{N}}. \end{cases} \tag{2.14}$$

Once more this naturally implies a subset of $\mathcal{N}'$ that $\mathcal{N}$ wants to work with as:

$$\mathcal{X}_{\mathcal{N}}(\mathcal{N}') = \{j \in \mathcal{N}' \ : \ r_{\mathcal{N}}(j) \leq \delta_{\mathcal{N}}\}. \tag{2.15}$$

Taking into account the fact that this can be arbitrarily decomposed to account for cases where there are multiple sub-networks that $\mathcal{N}$ wants to cooperate with, we are left with the following network-wide bifurcation of the network $\mathcal{N}'$:

$$\mathcal{N}' \xrightarrow{f_{\mathcal{N}}} \coprod_a \mathcal{X}^a_{\mathcal{N}} \coprod_b \bar{\mathcal{X}}^b_{\mathcal{N}}. \tag{2.16}$$

Note that $f_{\mathcal{N}}(\mathcal{N}') = 1$ implies that $\delta_i \leq \delta_{\mathcal{N}} \ \forall i \in \mathcal{N}$, which is the statement that if a network agrees to work with another, it means *all* individual nodes agree to do so and not just the majority. With these notions of distance and distrust in place, we can prescribe autonomous algorithms in which networks can freely bifurcate and recombine with the use of *abstract simplicial complexes* [2].

2.1.2 Network complexes

Before building a complex from a network $\mathcal{N}$, we must first motivate what [geometric/abstract] simplicial complexes are and what they can do for us. Consider first the linear dependence of points. For vectors $\vec{v}^a \in V$ in some vector space V, we say they are linearly independent if there exists coefficients $\lambda_a \in \mathbb{C}$ such that $\lambda_a \vec{v}^a \neq 0$. We extend this to what is known as *affine independence* if the following constraint — in addition to linear independence — holds true:

$$\sum_a \lambda_a \neq 0. \tag{2.17}$$

The intuition behind this is as follows. A set of vectors is linearly dependent if there are more vectors than necessary to generate their span. On the other hand, a set of vectors is affinely dependent if there are more vectors than necessary to generate their *affine hull*. To understand this, an *affine set* is the set which contains all affine combinations of points in it. For two points, the affine hull is an infinite line going through both points. On the other hand one could consider their *convex hull* [the smallest convex set[†] which contains it] would be the line segment which connects the two. The difference between the two types of hulls can be visualized in the following [3]:

[†]A set is convex if, given any two points in the subset, the subset contains the whole line segment that joins them.

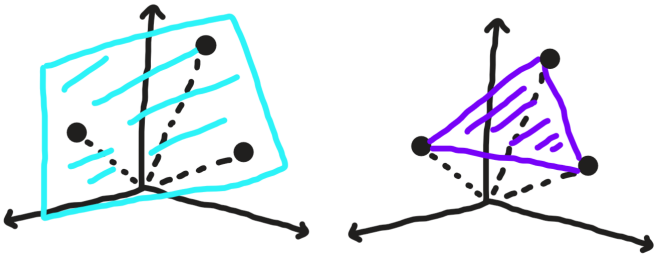


Figure 1. Visualization of different hulls for a set of embedded points. On the left [in blue] we have the affine hull which contains all possible affine combinations of the points. On the right [in purple] we have the convex hull which is the smallest convex set which contains the points.

Thus, for a set of $k+1$ affinely independent points $X \subset \mathbb{R}^d$, their convex hull — which is spanned by X — is known as its k-dimensional *simplex* σ. We say that the points or *vertices* of the simplex must be affinely independent to avoid the case where $k+1$ points trace out a simplex that is less than k dimensions. Moreover, simplices spanned by subsets of X are known as *faces*. We say a collection of a simplices is a *geometric simplicial complex* $\mathfrak{C}$ in $\mathbb{R}^d$ if the following hold:

i) Any face of a simplex of $\mathfrak{C}$ is in itself a simplex of $\mathfrak{C}$,

ii) The intersection of two simplices of $\mathfrak{C}$ is either empty or a common face of the two.

Given a geometric simplicial complex $\mathfrak{C}$ [a discrete object], we take the union of all simplices $\sigma \in \mathfrak{C}$ to form the *underlying space* of $\mathfrak{C}$, written as $\Xi(\mathfrak{C}) = \bigcup_a \sigma_a$ for $\sigma_a \in \mathfrak{C}$. The underlying space Ξ inherits its topology from the ambient space in which the simplicial complex resides in, which in our case is $\mathbb{R}^d$. From this perspective $\mathfrak{C}$ can be seen as a *continuous topological space* through Ξ. One notes that once all vertices are known [forming a vertex set V], $\mathfrak{C}$ is fully determined by a combinatoric description of simplices and incidence rules. From our description of geometric simplicial complexes, we can look at a its associated *abstract simplicial complex* $\tilde{\mathfrak{C}}$ which is constructed as follows:

$$\tilde{\mathfrak{C}} = \{\sigma \subset V : \sigma \supset \tau \subset V, V \ni v \subset V\}, \tag{2.18}$$

where τ is the *face* of the simplex σ. The first condition states that faces are also simplices, while the second states that all singletons [singular elements of V] are also simplices. The elements of $\tilde{\mathfrak{C}}$ is in essence a family of sets that is closed under taking subsets [every subset of a set in the family is also in the family]. We can have intuition regarding this condition by considering that in geometric simplicial complexes, all sub-simplices are also simplices in the set. One can always associate a topological space $|\tilde{\mathfrak{C}}|$ to an abstract simplicial complex $\tilde{\mathfrak{C}}$ such that if $\mathfrak{C}$ is a geometric complex whose combinatorial description is the same as $\tilde{\mathfrak{C}}$, the underlying space $\Xi(\mathfrak{C})$ is homeomorphic[†] to $|\tilde{\mathfrak{C}}|$. Abstract simplicial

[†]Bijective map between topological spaces that preserves topology. A class of homeomorphisms is precisely an equivalence class of the space with the same topological information.

complexes emphasize the underlying structure of a set of points via subset rules while geometric simplicial complexes emphasize how simplices are connected and its topological properties.

A particularly useful abstract simplicial complex is the *nerve complex* which records patterns of intersection of sets in a family of sets. Let I be an index set labelling sub-networks $\mathcal{N}_i$ of a network $\mathcal{N}$, meaning we have the decomposition $\mathcal{N} = \bigcup_{i\in I} \mathcal{N}_i$. These sub-networks are in fact over covers of the network and form an overall network cover written as $\mathcal{U} = \{\mathcal{N}_i\}_{i\in I}$. The nerve of the cover of the network is a set of [finite] index subsets $J \subseteq I$ such that the intersection of the covers $\mathcal{N}_{j\in J}$ is non-empty. We represent the nerve C of the network cover $\mathcal{U}(\mathcal{N})$ as the following:

$$C(\mathcal{U}) = \Big\{J \subseteq I : \bigcap_{j\in J} \mathcal{N}_j \neq \varnothing\Big\}. \tag{2.19}$$

This represents the set of vertices of the different sets which intersect and are joined by edges [lower dimensional simplices]. The construction of a nerve of a network given intersections of its sub-network covers is visualized through the following figure:

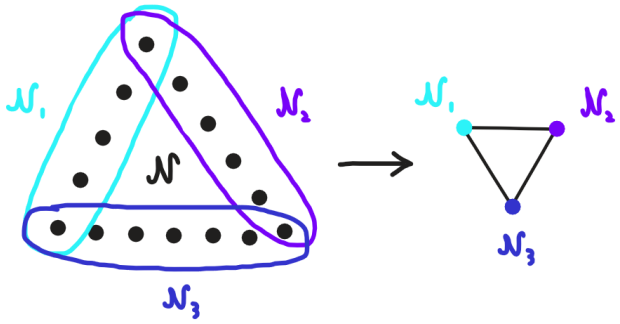


Figure 2. Nerve complex of a network $\mathcal{N}$ given a family of sub-network covers $\mathcal{N}_i$. The vertices correspond to the sets of the covers while the edges correspond to intersections between covers. The nerve complex gives information on patterns of intersection given a family of sets.

The nerve complex of the cover $\mathcal{U}$ of a network $\mathcal{N}$ is an abstract simplicial complex $C(\mathcal{U})$ whose vertices are the sub-networks $\mathcal{N}_i$. Whereas a k-simplex given a vertex set of points may be written as $\sigma = [x_1, \ldots, x_k]$, here we instead write $\sigma = [\mathcal{N}_1, \ldots \mathcal{N}_k]$. Now that we have a description of a nerve complex, we can relate the structure of an simplicial complex to its underlying topology [*which will be our main mechanism to work with continuous spaces given discrete network vertices*]. Specifically, we will relate the topology of a nerve of a network cover to the topology of the union of the sets of the cover through what is known as the *Nerve theorem*. Ideally we would want to relate the two through a homeomorphism [topology preserving mapping] mapping however it turns out to be too strong of a condition [2], so instead we require them to be related through *homotopy*. While homeomorphisms preserve all topological properties, homotopy focus on spaces being equivalent under *deformation* and preserving the topological properties which can be detected with computational methods. Moreover, it is much more straightforward to

demonstrate equivalence based on homotopy as opposed to homeomorphisms. Later we will characterize the topology of networks through their homology. If two spaces are homotopy equivalent, the induced maps on homology groups are identical. Therefore, homotopy equivalence serves as a suitable notion of equivalence for comparing spaces based on their homological properties. Now for the definition of homotopy.

Consider for example an equivalence class [for a primer on equivalence classes see 2.31] of maps which send points from an n-sphere S^n to a network $\mathcal{N}$, under the condition that a base point is preserved in the mapping. The equivalence class of these maps are known as *homotopy classes*. Two maps f_1, f_2 within this class are homotopically equivalent if they can be continuously deformed into one another via a continuous 1-parameter family of maps $f(t)$ which adhere to the conditions given by [4]:

$$f(t) : S^n \to \mathcal{N} \mid t \in [0,1],\ f(0) = f_1,\ f(1) = f_2 \tag{2.20}$$

This is to say the two spaces which f_1 and f_2 map to are homotopically equivalent and so can be continuously deformed into one another without passing through voids [roughly speaking these are higher dimensional holes]. The homotopy classes form a group structure known as the *n-th homotopy group*[†] of the network $\mathcal{N}$, denoted $\pi_n(\mathcal{N})$, which is the set of all homotopy classes of maps $f_k : S^n \to \mathcal{N}$ [5] for $k \in \mathbb{Z}$. The first homotopy group $\pi_1(\mathcal{N})$ is known as the fundamental group of a manifold, and is the group of equivalence classes under homotopy of loops within $\mathcal{N}$; that is to say the set of loops that can be continuously deformed into each other given a 1-parameter family of maps. Furthermore, the second homotopy group $\pi_2(\mathcal{N})$ is the group of homotopy classes of closed 2-dimensional surfaces in $\mathcal{N}$ [6] which can be deformed into one another. The intuition can be extended for compact n-dimensional manifolds corresponding to the n-th homology group $\pi_n(\mathcal{N})$. Furthermore, a space is said to be *contractible* is it is homotopy equivalent to a point. Homotopy groups are useful as they tell us about the topology of the network as it states what can be deformed into another given the *absence* of voids. Following the definitions of homotopy, we now present the **Nerve theorem** which allows us to relate the discrete topology of a network $\mathcal{N}$ to its continuous underlying space. Let $\mathcal{U} = \{\mathcal{N}_i\}_{i \in I}$ be a cover of a network $\mathcal{N}$ by open sub-network sets such that the intersection of any sub-collection of the $\mathcal{N}_i$ is either empty of contractible. Then, $\mathcal{N}$ and the nerve $C(\mathcal{U})$ are homotopy equivalent and thus share the similar topological features. The theorem is important as it encodes the topology of continuous spaces from abstract combinatorial structures [that are well suited for decomposition algorithms].

Now, how does one build simplicial complexes from a network $\mathcal{N}$ — a discrete collection of nodes that form a vertex set? Here we turn our attention to abstract simplicial complexes employed in topological data analysis which are known as the *Čhech complex* $\check{C}_\alpha(\mathcal{N})$, and the *Vietoris–Rips complex* $R_\alpha(\mathcal{N})$, both of which are constructed through the use of filtrations. A *filtration* $\mathcal{F}$ is a nested sequence of increasing subsets. Consider an

[†]It is noted that $\pi_n(\mathcal{N})$ is a group only for $n \geq 1$ and $\pi_0(\mathcal{N})$ is only a set. The non-triviality of homotopy groups give rise to what are known as *topological defects* which are often produced by spontaneous symmetry breaking events in high energy physics.

indexed set of sub-objects $\{S_i\}_{i\in I}$ of an algebraic structure S [group, vector field, set, etc.] such that: $S_i \subseteq S_j$ for $i \leq j$. We write this collection as $\mathcal{F} = \bigcup_{i\in I} S_i$. As the index decreases, the sub-objects get smaller thus giving us finer filters [or covers] to probe the topological properties of the space at different scales. The filtration we use to construct the aforementioned complexes are *offset* filtrations which involves taking an increasing sequence of balls centered at the nodes of our network and with this we take its nerve to trace out a simplicial complex [using the network as its vertex set]. Formally, for an offset filter on a network $\mathcal{N}$ where the radii of the balls are fixed to $\alpha \in \mathbb{R}$, we construct its simplicial complex by taking its nerve based on patterns of intersections of the ball filters. This creates the *Čhech complex* $\check{C}_\alpha(\mathcal{N})$ for our network $\mathcal{N}$. The criteria for tracing out simplices in the complex over the network is as follows:

i) If two circles mutually intersect, trace an edge between the vertices.

ii) If three circles mutually intersect, trace out a triangular simplex between the vertices.

iii) If four circles mutually intersect, trace out a polyhedral simplex between the vertices.

This can be extended to the higher dimensional case where for k-circles mutually intersecting [i.e. sharing a point of intersection], trace out a k-simplex whose vertices are the nodes the intersecting circles are centered at. A visualization of these rules can be seen in the following figure:

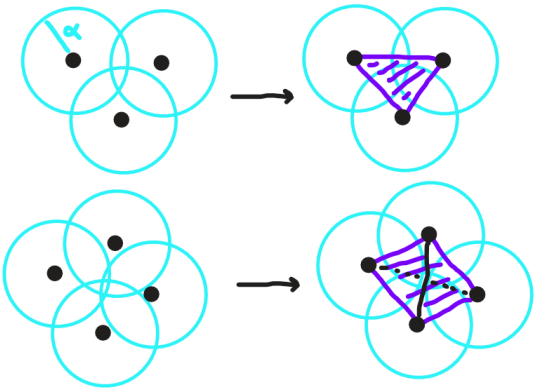


Figure 3. Construction of a Čhech complex given intersections of an offset filtration on a network [vertex set]. The top shows tracing out a planar triangle simplex for the mutual intersection of three circles while the bottom shows tracing out a polyhedron for the mutual intersection of four circles.

If the network $\mathcal{N}$ is a set of points in $\mathbb{R}^d$, the Čhech complex $\check{C}_\alpha(\mathcal{N})$ is homotopy equivalent to the union of the balls $\bigcup_{i\in\mathcal{N}} B(i,\alpha)$ and so the Čhech complex will always have a geometric realization in $\mathbb{R}^d$. In higher dimensions it becomes computationally difficult to deduce when n-balls have a point of intersection [where all of them mutually intersect]. For this reason we will instead turn the Vietoris-Rips complex $R_\alpha(\mathcal{N})$ given a network $\mathcal{N}$. In this case, instead of tracing out simplices based on conditions of the intersections

of the balls, it is instead based on the distance between vertices [or rather, the nodes of the network]. If two points have a separation that is smaller or equal to the α parameter [uniform radius of the filtration] then a corresponding simplex is traced out. Much like the Čhech complex, the dimension of the simplex is based on how many points mutually fall within the condition distance. The difference between the Čhech and Vietoris-Rips complex is visualized in the following figure:

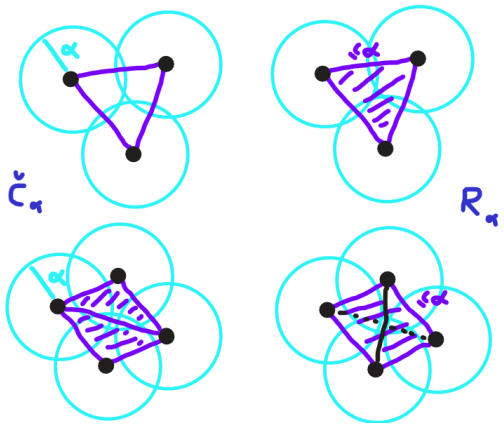


Figure 4. Difference between the Čhech and Vietoris-Rips complexes. The top left shows that each pair of circles has a point of intersection but not all circles simultaneously share a point of intersection, thus three edges are drawn and not a triangle. Top right on the other hand shows that being that all the points have separations less than or equal to α, a full triangle is traced out. Bottom left shows that the top three circles and the bottom three circles each have a point of intersection, however all four circles don't have a mutual point of intersection and so two triangles are traced out as opposed to a tetrahedron. The bottom right on the other hand traces out a full tetrahedron based on the conditions of the separations between the points.

Now that we have a prescription to trace out a simplicial complex for a given network — which emerges purely from the amount of trust all nodes have in each other — we move onto partitioning the network to maximize trust and scalability.

2.1.3 Trust-based network partitioning

Given a network $\mathcal{N}$ and its associated simplicial complex $\mathfrak{C}(\mathcal{N})$, how do we come up with decision outcomes that tells us how it will partition itself through bifurcations? Moreover, how do we modify the above to include the individual trust metrics of each node instead of a global filtration radius given by α? To build up to that, we first consider the case where a network complex is created via the offset filtration method mentioned above, meaning all nodes have the same trust in one another and so α is constant. For this, we have a geometric simplicial complex which is a collection of simplices σ_i, written as: $\mathfrak{C} = \{\sigma_i\}_{i\in I}$, for some labelling index set I. We can express this collection as disjoint union of simplices [recalling that vertices and edges are also simplices] and so naturally recover the decomposition:

$$\mathfrak{C} \longrightarrow \coprod_{i\in I} \sigma_i. \tag{2.21}$$

Clusters of nodes that form a local simplex of dimension of at least two are known as *cliques*. The constraints on the nodes being a part of a clique depends on what complex was created from the network [either Čhech or Vietoris-Rips filtrations]. Ideally we want to partition this complex in such a way that the network bifurcates into distinct simplices and this is done by cutting edges which connect higher dimensional simplices [$\dim \sigma > 1$]. After cutting, the network complex becomes a collection of complexes: $\mathfrak{C} \longrightarrow \{\mathfrak{C}'\}$ where $\mathfrak{C}' = \{\sigma'_j : \dim \sigma_j > 1, j \in J\}$, for some index set $J \subset I$ which labels the simplices of the new sub-complexes. The cutting of a network simplicial complex based on the dimensions of its simplices is visualized in the following:

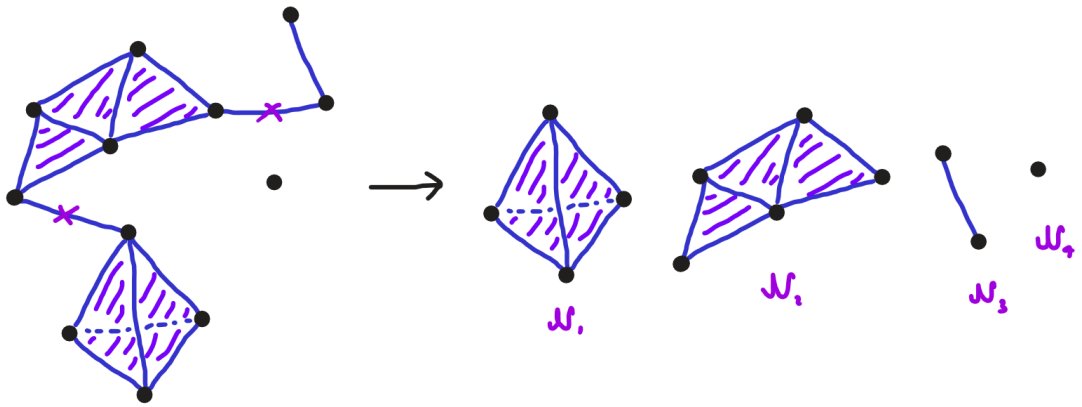


Figure 5. Partitioning of network simplicial complex via cuts on edges which connect higher dimensional simplices. The result is an independent collection of simplices which the network has been partitioned into.

We can write the partitioning or decomposition above of a network into four independent networks explicitly as the following mapping:

$$\mathfrak{C}(\mathcal{N}) \longrightarrow \{\mathfrak{C}'(\mathcal{N}_k) : k \in K\}, \tag{2.22}$$

for some index set K labelling the distribution of new simplicial complexes for the new networks. Now a caveat with the partitioning above of the simplicial complex of a consensus network through cuts is that for the case of network $\mathcal{N}_2$, all nodes don't trust each other such that their security parameters δ_i are individually satisfied [hence a bundle of triangles are formed instead of a higher dimensional polyhedron]. Thus, this clique cannot be contained within our decomposition following a partitioning. Furthermore, consider the case which is not taken into account when making cuts such as dimensional degeneracy:

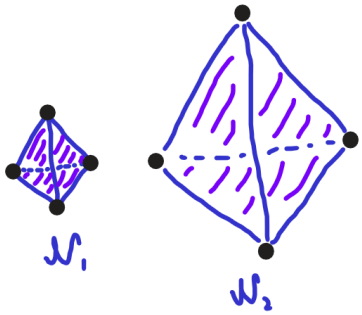


Figure 6. Dimensional degeneracy for simplices of the same dimension with different levels of trust. For the total amount of trust [as the summation of the trust of a node in the network for all nodes], it follows that: $r(\mathcal{N}_1) < r(\mathcal{N}_2)$. Recall that the distance between nodes represents the level of trust in each other.

The dimensionality alone does not capture the trust information of the clique and is disregarded for in the cutting prescription. Another piece of information not captured about our networking partitioning [in the case where all balls have the same radius $\alpha \in \mathbb{R}$] is the disparity between individual levels of trust that nodes have in each other. Given two nodes of a network $i, j \in \mathcal{N}$, it is not necessarily the case that $r_{ij} = r_{ji}$ and so the single edges that connect nodes does not capture this information:

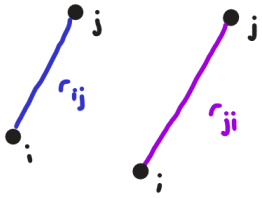


Figure 7. Disparity between the level of trust in two nodes, implying the lack of symmetry of the trust matrix $[r_{ij} \neq r_{ji}]$. This information could be captured by a generalization of the Čhech / Vietoris–Rips complexes in which the radius of the ball covers will be different for each nodes based on their amount of trust in one another.

With these issues we thus require a modification in the rules for our partitioning protocol for consensus networks, following the construction of their associated simplicial complexes. We propose a set of axioms for partitioning of a consensus network as the following:

i) *Higher dimensional simplices can only be connected by edges or be completely disconnected.*

ii) *A partitioning must maximize the total trust on the level of the simplicial complex and not just the dimensionality of its constituent simplices.*

iii) *An offset filtration must be generalized to have covers of different radius for each component of trust matrix of the network.*

We now explain the reasoning for each axiom and its relevance to our partitioning of consensus networks. Axiom (i) is to enforce the condition that $r_i(\mathcal{N}) \leq \delta_i$ for all cliques, meaning all nodes in the local clique simplex sufficiently trust each other — which for example was not the case for network $\mathcal{N}_2$ in Fig. 5. This in term produces only simplices of maximum dimension given maximum network trust. Axiom (ii) is to ensure that the trust is maximized over the network to select the best partitioning of the network — which was not the case for the dimensional degeneracy in Fig. 6. Axiom (iii) is to take into account the different individual levels of trust each node has in each other, which is captured by the trust matrix r_{ij} of the network — this is visualized in Fig. 7. In principle we could have a fourth axiom of the form:

iv) *The genus of the simplices of a network partitioning must be minimized.*

This is to avoid cases where a triple set of edges is formed instead of a triangular face given three nodes as this would not facilitate a network of maximal trust. However, it is not clear how to do this from the individual perspectives of the nodes given by r_{ij}, and on the global level this is covered in principle by axioms (i) and (ii). Henceforth, we should have a definition of the *total trust of a network* for the purposes of maximization [and subsequent optimization]:

$$r(\mathcal{N}) = \sum_{i \in \mathcal{N}} r_i(\mathcal{N}) = \frac{1}{|\mathcal{N}|} \sum_{i,j \in \mathcal{N}} r_{ij}(\mathcal{N}). \tag{2.23}$$

We will see in later sections on network evolutions, this is analogous to the total trust of a network history, given by $r(\mathcal{H})$ [and also is dependent on the topology of the network]. Furthermore, we require our offset filtration to be generalized to include the subjectivity of each node's perspective of trust. In the case before, given a network $\mathcal{N}$ as a collection of nodes, its offset filtration $\mathcal{F}(\mathcal{N})$ is a collection of balls centered at the nodes which can be expressed as $\mathcal{F}(\mathcal{N}) = \bigcup_{i \in \mathcal{N}} B(i, \alpha)$, where α is the filter parameter. For the subjectivity of trust amongst nodes in our model we instead allow the radius of a ball respective node's trust parameter δ_i. This gives what we call a *subjective filtration* which generalizes an offset filtration to $\mathcal{F}(\mathcal{N}) = \bigcup_{i \in \mathcal{N}} B(i, \delta_i)$. For tracing out a simplicial complex, we now look at the nerve of intersections of the subjective filtration and the subjectivity of the complex is characterized by the disparity of trust as seen in Fig. 7. Instead of tracing out an entire complex from the whole network, we will trace out a subjective complex based on the perspective of each node and this information encodes the structure of the total network complex [and hence how the system evolves]. We denote for a network $\mathcal{N}$, a *total network complex* as $\mathfrak{C}(\mathcal{N})$, as well as its *subjective node complexes* $\mathfrak{C}(i) \subset \mathfrak{C}(\mathcal{N})$ constituents. A visualization of the disparity of subjective complexes is presented in the following figure:

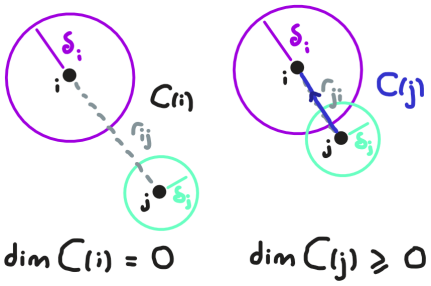


Figure 8. Disparity between subjective complexes constructed through taking the nerve complex of the subjective filtration. On the left we see the subjective complex $\mathfrak{C}(i)$ constructed from the perspective of i, while on the right we have $\mathfrak{C}(j)$. Notice an orientation on the edge formed, used to discern edges formed from different perspectives r_{ij} and r_{ji}, respectively.

From this construction we note that an individual node i has *only* information on how much it trusts other nodes — and so only the information of the i-th row of r_{ij} — as well as all the security parameters $\{\delta_i\}_{i\in\mathcal{N}}$ of the network $\mathcal{N}$. Being unable to know all the information of the network [meaning all the points of intersection of the network cover for a fixed perspective of r_{ij}], we see that the subjective nerve complex traced out can be at most a simplex of dimension one and thus an edge. This naturally gives us a dimension constraint on the complexes formed for different perspectives:

$$\dim \mathfrak{C}(\mathcal{N}) \geq 1 \geq \dim \mathfrak{C}(i), \tag{2.24}$$

which holds $\forall i \in \mathcal{N}$. How then do we look at the total simplicial complex for the network? This would be information we would need to partition the network accordingly. One could consider a function f which acts on the collection of the subjective complexes and from this construct a higher dimensional network complex. If for example we consider the case of three nodes such that $\mathcal{N} = \{i, j, k\}$ — thus three subjective complexes $\{\mathfrak{C}(i), \mathfrak{C}(j), \mathfrak{C}(k)\}$ — we denote this map as $f : \mathfrak{C}(i) \times \mathfrak{C}(j) \times \mathfrak{C}(k) \longrightarrow \mathfrak{C}(\mathcal{N})$. We can also denote this in general with the following representation:

$$f\Big(\mathfrak{C}(i) \sqcup \mathfrak{C}(j) \sqcup \mathfrak{C}(j)\Big) \longrightarrow \mathfrak{C}(\mathcal{N}). \tag{2.25}$$

This makes a decision on if the sets of complexes [which represent the trust as viewed from each node] form a total connected network complex. We visualize this combination in the following figure:

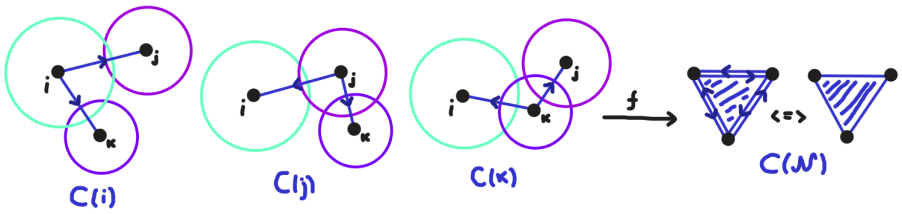


Figure 9. Construction of a total network complex given constituent subjective complexes, and a function f which maps them. A simple representation of the outcome is given following the action of f on $\{\mathfrak{C}(i) : i \in \mathcal{N}\}$.

Where does this arbitrary function f come from? Well we have seen it before to be represented by Eq. 2.9 in which based on conditions of the trust of a node in the network, it makes a decision to participate. While previously we defined the decision if a node wants to cooperate with a network as $f_i(\mathcal{N})$, we instead look at all decisions based on conditions of the individual trust matrix r_{ij}. Thus the action of the function f on the matrix r_{ij} gives us a boolean matrix $f(r_{ij}) \equiv f_{ij}$ which we define in the following:

$$f_{ij} = \begin{cases} 1, \, r_{ij} \leq \delta_i \\ 0, \, r_{ij} > \delta_i. \end{cases} \tag{2.26}$$

The number of pairs (f_{ij}, f_{ji}) such that $f_{ij} = f_{ji} = 1$ tells us the amount of edges in the oriented network complex $\mathfrak{C}(\mathcal{N})$. For all tuples $(f_{ij}, f_{ji}, f_{ik}, f_{ki}, f_{jk}, f_{kj})$ such that all of its elements equal one, this tells us how many triangular simplex faces we see in the network complex $\mathfrak{C}(\mathcal{N})$. In general the amount of k-tuples $(f_{ij}, \dots f_{ji})$ tells us how many q-simplices are formed [where $q = k(k+1)$] in the network complex $\mathfrak{C}(\mathcal{N})$. The different pairings of f_{ij} are in fact block matrices — sub-matrices or blocks which partition the matrix — of 1s which represent the different cliques formed as simplices. This can be visualized in the following components of the matrix f_{ij}:

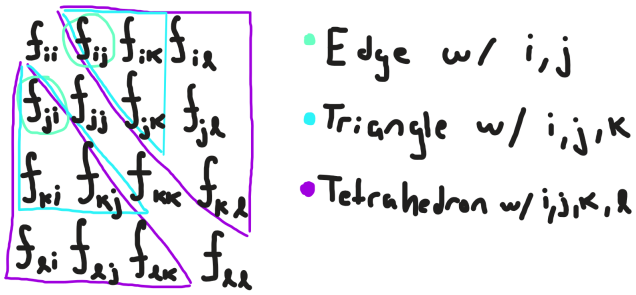


Figure 10. Block matrices forming simplices in the total network complex $\mathfrak{C}(\mathcal{N})$. We see that a pairing of two components (f_{ij}, f_{ji}) traces out an edge between nodes i and j given that $f_{ij} = f_{ji} = 1$. This is furthered by the formation of higher dimensional simplices given a respective pairing between an upper and lower triangular matrix of 1s which excludes the matrix diagonal [being that nodes are assumed to trust themselves completely].

The pairing of triangular components of f_{ij} [of unitary value] is precisely the condition

that for a mutual simplex to form, all nodes must have sufficient trust in one another. By looking at sub-triangular matrices of the boolean matrix f_{ij}, one can surmise the formed multi-oriented cliques of the network complex. Now given a node i, the only information it perceives are the values in its own row — that is to say $\{f_{ij}\}_{i\in\mathcal{N}}$. From its perspective, its in its interest to join a clique which maximizes the trust it has in others, as well as the clique dimension. It has no information on the total clique dimension as that information depends on the trust of the nodes independent to itself. Thus the *only* decision a node can make is to join based on the clique which maximizes its total trust, which is independent of size. For example consider a network $\mathcal{N} = \{i, j, k, l\}$ and their associated trust matrix r_{ij}, and the boolean matrix $f(r_{ij}) \equiv f_{ij}$. From the perspective of node i, it has the information of its row in r_{ij}, as well as its row in f_{ij}. For a given collection of $f_{ij} = 1$ in its row, we look at the corresponding values of r_{ij}. Consider the case where i trusts all other nodes such that $f_{ij} = f_{ik} = f_{il} = 1$, which means that $0 < r_{ij}, r_{ik}, r_{il} < \delta_i$. *How then does i pick the clique it wants to join?* This depends on the different values of r_{ij}. For the purposes of demonstration consider the case where $r_{ij} \leq r_{ik} \leq r_{il}$, meaning the node i trusts the most is node j. Naturally node i wants to work with j the most and would chose to form an edge with it to cooperate. For the purposes of computational power it would be in i's interest to form a larger clique even if they trust the other nodes slightly less compared to j. Node i will make a hierarchy of decisions and select the best from those options, prioritizing its first choice [and thereafter the second and third preference of clique]. The different selections are based on the total trust from the different possible cliques. The possible cliques that can be formed for the network $\mathcal{N}$ is given by its power set $\mathcal{P}(\mathcal{N})$[†]. To deduce the possible cliques that i can chose to be apart of we can restrict the power set to only those that include i — and the nodes that it trusts — which gives us the following: $\mathcal{P}(\mathcal{N})|_i = \Big\{\{i,j\},\{i,k\},\{i,l\},\{i,j,k\},\{i,j,l\},\{i,k,l\},\{i,j,k,l\}\Big\}$. For each clique, we look at the weighted trust by averaging over the trusts from the perspective of i, which for example we select the clique $\sigma = \{i, j, k, l\}$ and have the following:

$$r_i(\sigma) = \frac{1}{|\sigma|}\sum_{j\in\sigma} r_{ij} = \frac{1}{4}(r_{ij} + r_{ik} + r_{il}). \tag{2.27}$$

Thus the optimal choice for node i is the clique within the power set that minimizes the weighted trust sum $r_i(\sigma)$. If we label the different cliques in the power set with an index a, we say that $\mathcal{P}(\mathcal{N})|_i = \{\sigma_a : a \in A\}$ for some index set A. We can order this such that the most trust worthy clique appears first and we get: $\mathfrak{A}_i = \{\sigma_a : r_i(\sigma_a) < r_i(\sigma_{a+1})\}$. Naturally node i's highest preference of clique formed would be the first element in $\mathfrak{A}_i$, followed by its second choice and so forth. In order for the clique of highest preference [as seen by node i] to be formed, we also require that from the perspective of the other nodes in the clique that this is the most optional choice for them as well. If this is not the case, we loop through the remaining choices for its clique until it is satisfied. To do so we first define the intersection of ordered power sets as $\Omega = \{\!\{\mathfrak{A}_i \bigcap \mathfrak{A}_j : i, j \in \mathcal{N}\}\!\}$. Here we use the

[†]It is noted here that we exclude the elements of the power set that is the empty set and the sets of individual nodes. So implicitly what we actually mean is $\mathcal{P}(\mathcal{N}) \setminus \{\varnothing, i, j, k, l\}$.

double curly bracket notation to denote a *multiset* which can contain degenerate elements [which is the manifestation of an overlap in preference]. Once Ω is established, we look at the clique with the highest multiplicity in the set — that is the one that maximizes $\mu_\Omega(\sigma)$ — and form that as a clique for the system. If it turns out two elements have the same maximal multiplicity, we select the one with a higher dimension. This is somewhat unintuitively expressed as the following:

$$\tilde{\sigma} = \arg\max\left(\left\{\dim(\sigma) : \sigma \in \Omega,\ \mu_\Omega(\sigma) = \max_{\sigma' \in \Omega} \mu_\Omega(\sigma')\right\}\right). \tag{2.28}$$

Essentially the selected clique $\tilde{\sigma}$ maximizes over both multiplicity and dimension simultaneously. For example, in the case that $\mathcal{N} = \{i, j, k, l\}$ consider that the clique that satisfies the constraint above is $\tilde{\sigma} = \{i, j, k\}$. Following this we must remove all cliques in Ω that contain those nodes and in this case it leaves us with only $\{l\}$ as the only other clique that can be formed. In general we loop over the highest multiplicity cliques to form [keeping in mind to remove the elements with nodes in previously formed cliques $\tilde{\sigma}$] until Ω is empty. Once the dust settles we are left with a distribution of new networks given purely by trust preference amongst the nodes. We formulate this procedure with the following pseudocode algorithm:

Algorithm 1 Network partitioning through mutual trust preferences of nodes

function NETWORKPARTITION($\mathcal{N}, r_{ij}, \{\delta_i\}$) $\longrightarrow \{\mathcal{N}\}$

- ▷ *Constructing the network complex with subjective filtrations* ◁
- **for** $i \in \mathcal{N}$ **do**
  - $i \xrightarrow{\mathcal{F}} \mathfrak{C}(i)$
- $\{\mathfrak{C}(i)\} \xrightarrow{f} \mathfrak{C}(\mathcal{N})$
- ▷ *Constructing the ordered cliques for each node* ◁
- **for** $i \in \mathcal{N}$ **do**
  - $i \longrightarrow \mathcal{P}(\mathcal{N})|_i \longrightarrow \mathfrak{A}_i$
- ▷ *Construction of intersections of clique preferences* ◁
- **for** $i, j \in \mathcal{N}$ **do**
  - $\Omega \longrightarrow \{\!\{\mathfrak{A}_i \bigcap \mathfrak{A}_j : i, j \in \mathcal{N}\}\!\}$
- ▷ *Construction of cliques σ given maximal mutual preferences* ◁
- **while** $\Omega \neq \varnothing$ **do**
  - $\tilde{\sigma} \longrightarrow \arg\max\left(\{\dim(\sigma) : \sigma \in \Omega,\ \mu_\Omega(\sigma) = \max_{\sigma' \in \Omega} \mu_\Omega(\sigma')\}\right)$
  - $\Omega \longrightarrow \Omega \setminus \left(\{\tilde{\sigma}\} \cup \{\sigma : \sigma \ni i \in \tilde{\sigma}\}\right)$
- ▷ *Defining the formed cliques as new respective networks* ◁
- $\{\tilde{\sigma}\} \longrightarrow \{\mathcal{N}\}$
- ▷ *Returning a new distribution of consensus networks* ◁
- **return** $\{\mathcal{N}\}$

A problem however with the algorithm above is that for a network of size N, a standard result in combinatorics is that there are 2^N different partitionings of the set [given

through its power set] which means the algorithm has *exponential complexity*. This makes is unfeasible to perform as one scales up N in which looping over all possible clique partitioning quickly diverges in the amount of time required to do so. What's more is that in constructing the largest mutually agreed on clique, one needs to have central oversight as nodes do not have the information of trust from one another. We require an algorithm that is purely subjective based on greedy decisions and one in which the *emergent* behaviour is global. Thus we require something with linear or polynomial in time complexity for computational feasibility, as well as a protocol that is purely locally subjective [and whose collective behaviour is an emergent global phenomena].

Thus, instead of having a network partition itself into arbitrary constituent pieces, we will instead consider many networks $\{\mathcal{N}\}$ and see how that distribution changes over time with trust-based decisions. Each node in each network will make a decision on which network it wants to join, whether that is its own network, another network, or none at all [in which case it abandons its own network to join an empty set and form its own net network]. The time complexity of such an algorithm is *linear* and scales based on how many total nodes there are in all networks combined. We represent this linear protocol through the following algorithm:

Algorithm 2 Linear network partitioning in the presence of external networks

function NetworkPartition($\{\mathcal{N}\}, r_{ij}, \{\delta_i\}) \longrightarrow \{\mathcal{N}\}$

- ▷ *Construction of optimal locations γ_i for each node* ◁
- **for** $i \in \mathcal{N} \in \{\mathcal{N}\}$ **do**
  - $\mathfrak{A}_i \longrightarrow \{\sigma_a \in \{\mathcal{N}\} : r_i(\sigma_a) < r_i(\sigma_{a+1})\}$
  - $\gamma_i \longrightarrow \arg\min_{\sigma_a \in \mathfrak{A}_i} r_i(\sigma_a)$
- ▷ *Performing node jumps given sufficient trust* ◁
- **for** $i \in \mathcal{N} \in \{\mathcal{N}\}$ **do**
  - **if** $r_{\gamma_i}(i) < \delta_{\gamma_i}$ **then**
    - $i \longrightarrow \gamma_i \cup \{i\}$
    - $\mathcal{N} \longrightarrow \mathcal{N} \setminus \{i\}$
- ▷ *Performing node abandons given insufficient trust* ◁
- **for** $i \in \mathcal{N} \in \{\mathcal{N}\}$ **do**
  - **if** $r_i(\mathcal{N}) > \delta_i$ **then**
    - $i \longrightarrow \varnothing_i \cup \{i\}$
    - $\mathcal{N} \longrightarrow \mathcal{N} \setminus \{i\}$
- ▷ *Returning a new distribution of consensus networks* ◁
- **return** $\{\mathcal{N}\}$

The linear algorithm above which is based on greedy decisions may be visualized through the following figure:

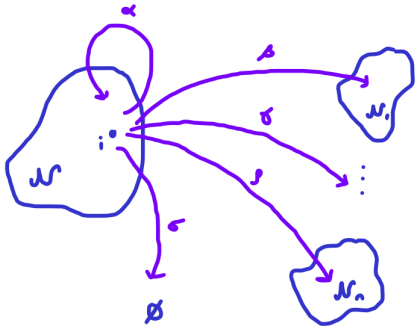


Figure 11. All possibilities of decision which can be greedily made from the perspective of node $i \in \mathcal{N}$. Each of the possibilities $\{\alpha, \ldots, \sigma\}$ is weighted by a probability based on the trust the node has on the corresponding network and will be later implemented in a topological path integral approach.

We implement this algorithm in Python [for details see appendix A] which can take as input an arbitrary amount of networks — as well as their associated trust matrices and security parameters — and output an optimal redistribution or partitioning of the networks. As an example consider the nodes $\{i, j, k, l, m, n, o, p, q, r\}$ — for coding purposes will be respectively labelled as $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ — distributed amongst four networks as: $A = \{0, 1, 2\}, B = \{3, 4\}$, $C = \{5, 6, 7\}$, and $D = \{8, 9\}$. Given certain values of trust and security parameters, the output of the partitioning algorithm looks like the following:

```
--- Original state of networks ---
Initial network A : [0, 1, 2]
Initial network B : [3, 4]
Initial network C : [5, 6, 7]
Initial network D : [8, 9]
Full optimal locations: [[1, 1, 1], [3, 3], [2, 2, 2], [0, 3]]
Node 0 in network A wants to jump network B.
Node 0 has jumped to network B.
Node 1 in network A wants to jump network B.
Node 1 has jumped to network B.
Node 2 in network A wants to jump network B.
Node 2 has jumped to network B.
Node 3 in network B wants to jump network D.
Node 4 in network B wants to jump network D.
Node 8 in network D wants to jump network A.
Networks after jumping events: [[], [3, 4, 0, 1, 2], [5, 6, 7], [8, 9]]
Node 4 in network B has abandoned its network.
Node 6 in network C has abandoned its network.
Node 8 in network D has abandoned its network.
Node 9 in network D has abandoned its network.
Networks after abandon events: [[], [3, 0, 1, 2], [5, 7], []]
```

```
Emtpy sets after evolution: {4: [4], 6: [6], 8: [8], 9: [9]}
Updated networks: [[0, 1, 2, 3], [5, 7], [4], [6], [8], [9]]
```

We may visualize the outcome of the algorithm acting on the distribution of networks through the following figure:

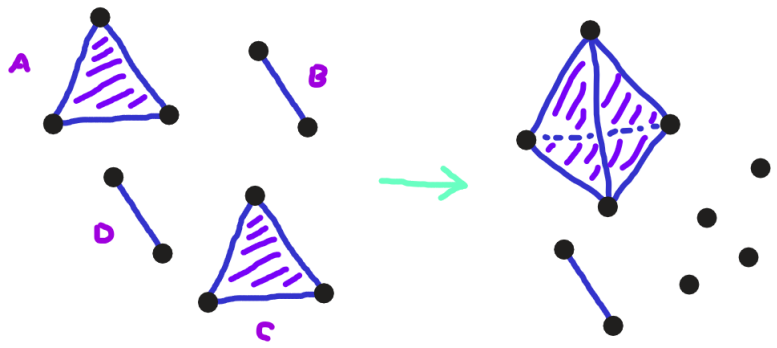


Figure 12. Partitioning algorithm executed on an initial distribution of consensus networks. Prior to the execution the maximum clique dimension was two whereas after it is four being that this configuration maximizes the amount of trust over the whole set of networks.

The algorithm gives us a prescription of a distribution of networks partitioning themselves into another distribution of networks based solely on conditions of trust nodes have in one another. Each execution of the algorithm represents a discrete time step t and may be iterated to trace out a discrete evolution of the network. If we denote the total evolution of many iterations of the algorithm on the network as a *network history* $\mathcal{H}$, then a network at a given time step is denoted as $\mathcal{H}_t$. In this sense naively we take $\mathcal{H} = \bigcup_t \mathcal{H}_t$ gives us our discrete network history. Although the structure of the network is discrete given simplicial network complexes, we can look at its continuous evolution by considering is underlying space. In this sense by iterating the algorithm we continuously trace out a network history [analogous to tracing out a spacetime history when evolving a gravitational system] which we denote as $\mathcal{H}$. We will see that it is less trivial than just taking the union of time slices and instead the full network history is created by gluing cobordisms which represents the continuous evolution of networks. We consider this in the next sections where our consensus networks are generalized to *topological consensus networks*. Whereas our analysis here on partitioning networks gives us information at [instantaneous] finer timescales, we will see in the next section that looking at the information of histories $\mathcal{H}$ gives us a coarser scale picture of what is going on. In the future we will make use of the network complexes to study the topology at the scale of instantaneous networks, while in the next section by studying the topology of network histories, we get global information of the complete evolution of the networks. The study of topology at different grained scales is characterized by *persistent homology*, a field of mathematics that is widely employed in topological data analysis.

2.2 Topological consensus networks

Here we outline the topological characterization of consensus networks, a necessity to study the complete continuous evolution of a consensus network. First we look at how a topological consensus network — described as a manifold — can bifurcate into different subnetworks via compliance checks on trust at the end of consensus events. We demonstrate the different ways they may evolve in time with the use of cobordisms to represent their temporal history. With the use of topological invariants — such as those arising in knot theory — we characterize a class of consensus network histories to determine how networks may autonomously combine with other networks based on criteria set by the network and clients. Finally, we demonstrate how the intersection of trust shared by the network and client can be used as a service.

2.2.1 Network history as cobordisms

Let's begin with the construction of a topological consensus network. A *regular* consensus network — whose elements are referred to as *nodes* — is a closed 1D set of parties wanting to have their decentralized transactions approved. To do so, for each transaction a node requests to be verified, so too must they participate in an equal amount of consensus rounds of transaction verifications. To use a *topological* theory, we require more dimensions. This is due to 1D having relatively trivial topology, with all loops in it contracting to a single point without encountering discontinuities. This comes from a 1D theory lacking *voids*[†] which are higher dimensional empty sets which cause the space to have non-zero genus [roughly speaking the number of holes]. For a 2D theory we will consider a space which contains the information of all possible nodes and transactions, and define it as the product space:

$$\mathcal{Y} \equiv \mathcal{N} \times \mathcal{T}, \tag{2.29}$$

where $\mathcal{N}$ is the set of all possible nodes and $\mathcal{T}$ is the set of all possible transactions. We refer to $\mathcal{Y}$ — the space of all nodes and transactions — as the *ambient* space. Being that the information of the nodes and their transactions are independent [and discrete] from one another we could just take $(\mathcal{N}, \mathcal{T}) = (\mathbb{Z}, \mathbb{Z})$ thus $\mathcal{Y} = \mathbb{Z}^2$. However, we will utilize the framework of a *continuous* theory [to define the notion of continuous functionals and operators] by taking the continuum limit and replacing the sets to $(\mathcal{N}, \mathcal{T}) = (\mathbb{R}, \mathbb{R})$ and so we instead have $\mathcal{Y} = \mathbb{R}^2$. To have a *topological* space, we require that the set $\mathcal{Y}$ and the collection of its *open* subsets satisfy the conditions [7]:

i) $\mathcal{Y}$ and its empty subset are open,

ii) If subsets $(\mathcal{Y}_a, \mathcal{Y}_b) \subseteq \mathcal{Y}$ are open, this implies the intersection $\mathcal{Y}_a \cap \mathcal{Y}_b$ is also open,

iii) If subsets $\mathcal{Y}_a \subseteq \mathcal{Y}$ are open, this means the union space $\bigcup_a \mathcal{Y}_a$ is also open.

[†]For some space $\mathcal{M}$ with $\dim \mathcal{M} > 1$, we can consider the Betti number $b_2 = \dim H^2(\mathcal{M})$ where $H^2(\mathcal{M})$ is the second cohomology group over $\mathcal{M}$. It tells us which 2-forms exist in the cotangent space over $\mathcal{M}$ as well as how many voids [or cavities] there are.

The open subsets $\mathcal{Y}_a$ form the *topology* of $\mathcal{Y}$ and cover the entire topological space. On each of these subsets we define a *chart* to be a continuous map $\varphi_a : \mathcal{Y}_a \to \mathbb{R}^2$. These charts allow us to work in locally flat coordinates [whereby the connection on the space of 2-forms, $\Lambda^2\mathcal{Y}$, vanishes]. Finally, to be a *topological manifold*, for any two charts (φ_a, φ_b) on $\mathcal{Y}$, the *transition function* which is written as the composition $\varphi_a \circ \varphi_b^{-1}$ must be smooth [integrable] over $\mathcal{Y}$. We thus define a **topological consensus network** $\mathcal{G}$ as a submanifold of the product space $\mathcal{Y}$ that is closed, meaning it contains it's boundary [for $\mathcal{G} \subseteq \mathcal{Y}$ and $\dim \partial\mathcal{G} = \dim \mathcal{Y} - 1$, then $\partial\mathcal{G} \neq \varnothing$].

Now, one could imagine how such networks can evolve over time. We can capture this information succinctly with the use of *Lorentzian*[†] manifolds such as that of spacetime in general relativity. If $\mathcal{Y}$ contains all possible information of the network's nodes and transactions, we can consider all possible evolutions of the network through the inclusion of an extra temporal dimension which we take to be simply $\mathbb{R}$. The inclusion of time is done through a Cartesian product and thus we define our **networktime** as the following product manifold:

$$\mathcal{M} = \mathbb{R} \times \mathcal{Y}. \tag{2.30}$$

Here $\mathbb{R}$ is the real line giving us a natural parametrization for time, and $\mathcal{Y}$ is a 2D manifolds describing the space of topological consensus networks. By this construction $\mathcal{M}$ is the space of all possible topological consensus network outcomes. *What does an evolution of a topological consensus network $\mathcal{G}$ in $\mathcal{M}$ look like?* Consider some topological consensus network $\mathcal{G} \subseteq \mathcal{Y}$ which is invariant under time translations [does not evolve in time]. Such an evolution would trivially trace out the following manifold in $\mathcal{M}$ as:

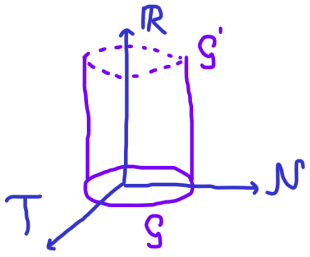


Figure 13. Evolution of a static topological consensus network $\mathcal{G}$ in the ambient networktime space $\mathcal{M} = \mathbb{R} \times (\mathcal{N} \times \mathcal{T})$. The network evolves trivially to the same network $\mathcal{G}'$ some time later and traces out a cylindrical manifold in networktime $\mathcal{M}$.

At first glance time evolution appears to be specified through boundary conditions on the network along the time dimension, separated by some time interval $\mathcal{I} \subset \mathbb{R}$. The

[†][Pseudo-Riemannian] manifolds with a metric signature of $(1, n-1)$, meaning its eigenvalues signs are given by a list $(-, +, \ldots, +)$. Such is the case for the *Minkowski* space metric $\eta_{\mu\nu} = \text{diag}(-1, +1, +1, +1)$ in general relativity when $D = 4$. The first eigenvalue corresponds to the time dimension, and due to the negative sign, measuring distances in Minkowski space depends on the following line element:

$$ds^2 = \eta_{ab} dx^a \otimes dx^b = -dt^2 + dx^2 + dy^2 + dz^2.$$

manifold traced out can be described via the product $\mathcal{H} = \mathcal{G} \times \mathcal{I} \subset \mathcal{M}$. Here $\mathcal{H}$ is the **network history** which captures all information of the network over time. It is utilizing these histories which will allow us to determine compatibility of independent networks to autonomously combine. A subtlety must be noted which is that there is a *degeneracy* in the history of a network for a given identical boundary condition; the same conditions can apply to different evolutions of a network. Consider two cases: one in which the initial and boundary condition is specified to be a single network, and one in which only the boundary condition is specified to be a single network. For these two cases we have the following scenarios:

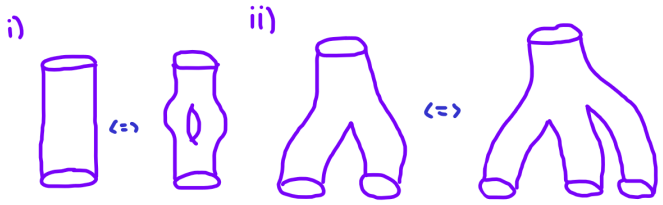


Figure 14. Visualization of boundary condition degeneracy. On the left subfigure we see specifying the same initial and boundary conditions can admit multiple unique histories and so there is a *degeneracy* [or multiplicity]. On the right subfigure we see that only specifying the boundary condition can have multiple histories satisfying that condition, again implying degeneracy.

While the second subfigure demonstrates the equivalence of histories of different topological consensus networks combining to form a single network [for different amounts of initial networks], the first subfigure is a bit strange; one of the histories has non-trivial genus[†]. How do we interpret this? The history admitting a void can be decomposed as the following:

Figure 15. Decomposition of a topological consensus network history with non-trivial genus. Here we have split it into two other histories and identified different points on the boundaries. The splitting of a manifold this way is known as a surgery [in surgery theory], whereas combining the manifolds is known as gluing. More on this later.

There are two ways to interpret this decomposition. On one hand the network starts as a whole, goes through a round of consensus and bifurcates into two networks due to suspected malicious nodes. Then after another round of consensus where the malicious

[†]Formally, the genus of a surface [with no punctures] is the maximum number of non-self-intersecting closed curves that you can draw on a surface without disconnecting it. While a torus has a unity genus, a pair of pants for example has vanishing genus.

nature is proven to be minimal, it recombines into a single network. The other way of viewing this decomposition is combining one network history with another, i.e., a method in which networks can combine to form larger networks for more efficient communication and verification of transactions.

The degeneracy as noted before — either coming from specifying one or both conditions — means that to describe a network history, it isn't enough simply to specify both the initial and final conditions. One must also take into account the information between these conditions, and to do so we will make use of cobordisms[†] and surgery theory. To do so we first review equivalence relations and classes. An *equivalence class* is a set where all the elements are equivalent to each other in some way. Given a set X and some element $a \in X$, the equivalence class of a in X is given by:

$$[a] = \{x \in X : x \sim a\}. \tag{2.31}$$

Here $\sim$ is the *equivalence relation* which tells us how two elements are equivalent [the most common equivalence relation is the equal symbol: $=$]. Thus, the class $[a]$ is the set of elements of X that are equivalent to a, and is known as a *partition* of X. The set of all equivalence classes or partitions of X is known as the *quotient set*, which is defined as:

$$X/\sim = \{[x] : x \in X\}. \tag{2.32}$$

The quotient set is usually defined between two sets, such as X/Y for some other set Y. This specifies the equivalence relation where two elements of a partition of X are equivalent if they differ by an element of Y. We will see that the specification to uniquely define a network history comes from a combinatoric series of possible network interactions. Now, a *cobordism* is equivalence relation on the class of compact manifolds of the same dimension, and can be interpreted as a set of instructions for a manifold M to evolve over time into another manifold N in such a way that their own boundaries are preserved [8]. Put more precisely, an $(n+1)$-dimensional cobordism is a quintuple of information:

$$(W; M, N, i, j), \tag{2.33}$$

where W is an $(n+1)$-dimensional compact manifold with a boundary $[\partial W]$, (M, N) are compact n-manifolds, and $i : M \hookrightarrow \partial W$ and $j : N \hookrightarrow \partial W$ are embeddings [structure contained within another instance, such as a subgroup within a group] of the manifolds to the boundary of W. The embeddings have disjoint images which adhere to the union condition $i(M) \sqcup j(N) = \partial W$. We can visualize this cobordism via the following figure:

[†]Cobordisms are used to describe wormholes and entangled particle creation in *topological quantum field theories*. In such theories, one considers a category in which the *objects* are compact n-dimensional manifolds and the *morphisms* [the maps between the objects] are cobordisms. Topological defects on these objects are interpreted as particles, and in this sense we can understand particle interaction through different mappings of the category's objects.

Figure 16. Cobordism between two compact manifolds (M, N) via an intermediary manifold W. Here the boundary of W is the disjoint union of M and N, given as $\partial W = M \sqcup N$. Here we have dropped the embedding maps as it is implied the compact manifolds (M, N) are contained within the *closure* of the higher dimensional manifold W.

Two manifolds (M, N) are called *cobordant* if such a cobordism exists and they share topological properties such as Chern/Pontryagin numbers [9]. From this we can construct *cobordism classes* which consist of all manifolds that are cobordant to a fixed manifold. In this sense we can fix the initial compact manifold [our topological consensus network] and consider the cobordism class of all manifolds which it is cobordant to, meaning all possible outcomes of the evolution of the network. More explicitly, two cobordisms of networks in this class are considered equivalent if they can be continuously deformed into each other [while homotopy deals with continuous deformations within the same space, cobordisms extend this idea to transitions between different spaces]. Thus, cobordisms give us a prescription on how to characterize network bifurcation based on *trust* evolving overtime with each round of consensus [and subsequent compliance checks]. The cobordism manifolds correspond precisely to the *history* $\mathcal{H}$ of a topological network $\mathcal{G}$ and hence how it evolves in time.

Now for some intuitions on cobordisms [of topological consensus networks], let us consider a few examples. The most trivial example of a cobordism is the 1D cobordism between 0D manifolds $M = \{0\}$ and $N = \{1\}$, which is given by the unit interval $W = [0, 1]$. Another fairly simple example would in in Fig. 15 where the first piece of the decomposition represents a cobordism between some initial topological consensus network $\mathcal{G}$ and two temporary networks. Finally, consider a cobordism from a 2-disk D^2 [filled in unit circle S^1] to a modified 2-disk Y with a handle H attached [8]. We can write such a cobordism as a mapping $\alpha : D^2 \Rightarrow Y \circ H$ and visualize it as the following:

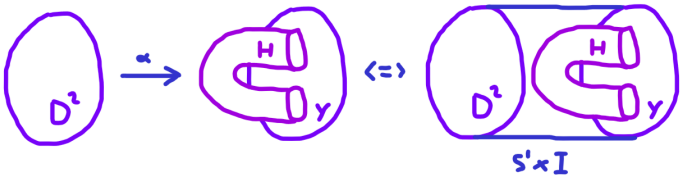


Figure 17. Cobordism between a disk and a modified disk with a handle. Here we characterize the length of the boundary of the cobordism [the distance between the cobordant manifolds] by some interval I and so write the boundary as $\partial W = S^1 \times I = D^2 \sqcup Y \circ H$.

Thus we can interpret the [categorical][†] cobordism as a continuous mapping between

[†]Here α in in fact a 2-morphism in a 2-category. This cobordism can be understood as a pair creation of

compact manifolds whose boundary is given by a higher dimensional ambient manifold. This is a direct result of the relative **trust** of nodes in a given network and is why the topological network can bifurcate into sub-networks in the first place. Thus, we will use cobordisms as the language to describe the evolution of a topological consensus network, which is succinctly captured by the network history in networktime. In the following section we will describe how different cobordisms [and hence network histories] maybe be combined via surgeries in a combinatoric sense, analogous to particle interactions in quantum field theory.

2.2.2 Combinatoric network scaling

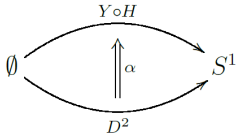
Now that we have a prescription for how topological consensus networks bifurcate into sub-networks due to breaches in trust [via cobordisms], we discuss how separate networks can *interact* with each other. An equivalent statement would be how independent network histories can *combine* into composite networks. Recall that boundary conditions alone are not enough to specify a unique network evolution and so we require additional information. We characterize this missing information as the interaction of different network histories [cobordisms] via a series of possible network interactions.

First, we will make use of *surgery theory* to make sense of stitching together different networks which have evolved independently in time. The purpose of such a theory is to produce a finite dimensional manifold from one or many others in a well-defined manner. A *surgery* refers to cutting out parts of a manifold and replacing it with a part of another manifold in such a way that its topological invariants are preserved. The cut is matched up along open boundary subsets of the manifolds [given the existence of a diffeomorphism ϕ between them]. Morse theory tells us that a manifold can be obtained from a surgery by a sequence of spherical modifications if and only if the manifolds belong to the same cobordism [equivalence] class. Thus we can combine the network histories under a *gluing* surgery in which the open subsets are identified [the points on each subset must be the exact same] if the histories are in the **same** cobordism class. We present gluing surgery in the following.

Consider two manifolds (M, N) with respective open subsets (U, V) which we would like to glue along by identifying the points of each subset. To do so, we require the existence of the diffeomorphism $\phi : U \to V$ between subsets. We write the enlarged manifold [M glued with N] as the space [10]:

$$M \cup_\phi N = (M \sqcup N)/\sim . \tag{2.34}$$

particles in a topological quantum field theory. Being that D^2 and $Y \circ H$ can be viewed as cobordisms from the empty set to the unit circle S^1, we can capture the cobordism in the figure via the following categorical diagram [8]:



Here we say that the two manifolds glued together is the disjoint union of the two with an identification of open subsets [hence the modulo equivalence relation]. This is the set of all equivalence classes or partitions of $M \sqcup N$ such that two elements $u \in U$, $v \in V$ are equivalent up to the diffeomorphism ϕ: $u \sim v = \phi(u)$. This means that the points exist on the same glued boundary. *What would this look like for network histories part of the same cobordism class?* Consider two network histories $(\mathcal{H}, \mathcal{H}')$ with respective boundaries $(\partial\mathcal{H}, \partial\mathcal{H}')$. We interpret these boundaries as a disjoint collection of initial and final topological consensus networks [that is to say, the amount of independent networks before and after time evolution]. For example, a network which splits in two will have a network history with three disjoint pieces: one initial network piece and two final network pieces. One could generally consider a network history $\mathcal{H}$ with n initial networks $\{\mathcal{G}_a\}$ and m output networks $\{\tilde{\mathcal{G}}_b\}$. We would write such a boundary of a the history as:

$$\partial\mathcal{H} = \coprod_{a=1}^{n} \mathcal{G}_a \coprod_{b=1}^{m} \tilde{\mathcal{G}}_b. \tag{2.35}$$

Thus for two network histories $(\mathcal{H}, \mathcal{H}')$ to be glued, we require the existence of diffeomorphisms that maps between the output boundaries of $\mathcal{H}$ to the input boundaries of $\mathcal{H}'$. If $\partial\mathcal{H}$ contains output boundaries $\tilde{\mathcal{G}}_a$ and $\partial\mathcal{H}'$ contains input boundaries $\mathcal{G}'_b$, then to glue the histories together we require the existence of the diffeomorphisms:

$$\phi_{ab} : \tilde{\mathcal{G}}_a \to \mathcal{G}'_b. \tag{2.36}$$

For the resulting glued space to remain a manifold, we require the graph of ϕ_{ab} — given by $\Gamma(\phi_{ab}) = \{(h, \phi_{ab}(h)) \mid h \in \mathcal{H}\} \subset \mathcal{H} \times \mathcal{H}'$ — to be closed in the product $\mathcal{H} \times \mathcal{H}'$. This ensures that the resulting space is Hausdorff and thus a manifold. Finally with the set of diffeomorphisms defined as $\{\phi_{ab}\} \equiv \phi$, we write down the **glued topological consensus network history** as:

$$\mathcal{H} \cup_\phi \mathcal{H}' = (\mathcal{H} \sqcup \mathcal{H}')/\sim, \tag{2.37}$$

where elements of the glued space are equivalent up to the diffeomorphisms ϕ [that map between the disconnected network pieces of $\mathcal{H}$ and $\mathcal{H}'$]. Now that we know how to combine network histories, we must consider how networks can *interact*. That is to say, what are the different ways networks can interact given a set of initial and boundary conditions? For this we take inspiration of the combinatorics behind particle interaction in *quantum field theory* (QFT). Consider a *scalar* quantum field theory, meaning a quantum theory that contains a scalar field representing bosons of spin 0. We encode how different bosons can interact given a vertex term $\lambda\varphi^4$ in the theory's Lagrangian $\mathcal{L}$, which looks like:

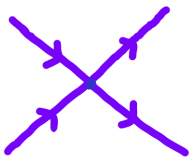


Figure 18. Interaction vertex of φ^4 scalar quantum field theory that allows for two incoming and outgoing particles. The vertex can be iteratively combined with other vertices to come up with potential particle evolutions through interaction.

Now given this interaction vertex we can describe particle *interactions*, otherwise known as the mechanism through which particles can fundamentally share information. All the different ways in which they share information — or *interact* — is given by all possible permutations of connecting interaction vertices. For example, can a single particle interact with itself and if so, what are all the ways it can? We visualize a spin 0 bosonic particle interacting with itself with the use of 2-point functions [expectation values of two operators $\langle\varphi\varphi\rangle$ in statistical mechanics when rotating our system to imaginary time] :

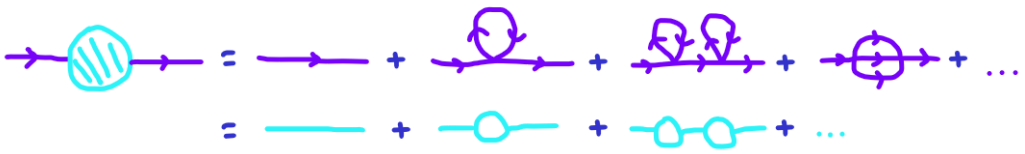


Figure 19. 2-point function for scalar particles. The first equality represents all possible ways a particle can come in, interact in some way, and then outcome as a single particle. The amplitude of a diagram [squared] in the series is the probability of the particle undergoing said diagramatic evolution. The second equality groups up all diagrams into tree-level [no loops] and different loop level diagrams.

This encodes all possible ways the particle can interact with itself over time, thus all possible particle evolutions. We can also consider all possible ways two incoming particles interact in some way and outcome as two particles. This is captured by the 4-point function in quantum field theory [expectation value of four operators $\langle\varphi^4\rangle$ this time]:

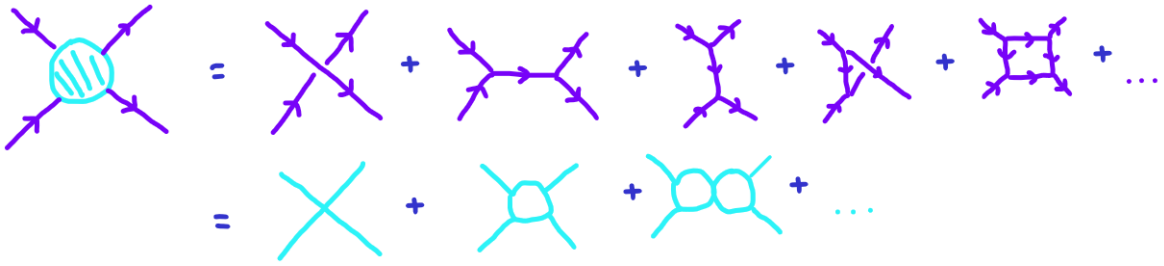


Figure 20. 4-point function for scalar particles. This represents all possible ways two particles can come in, interact in some way, and then outcome as two particles.

The different diagrams are given by permutations of combining different interaction vertices. Thus, an n-point function is a combinatoric expansion of $n/2$ particles interacting with each other as an addition of all possible diagrams. *Now what about the case of topological consensus networks?* We would like to encode all possible ways a network can evolve in time and interact with other networks given a fixed cobordism class Ω and boundary conditions $\mathcal{B}$. Analogously to quantum field theory, we would like an interaction vertex to prescribe how network histories may interact via permutations of gluing surgeries. We define such an interaction vertex for the topological consensus network as the 3-prong:

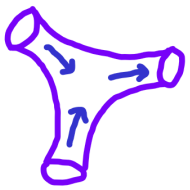


Figure 21. Topological interaction vertex of consensus networks, analogous to the vertex from φ^3 theory. From this we can combine it with other topological vertices to come up with all possible evolutions of a network, otherwise meaning all possible topological network histories in networktime.

With this we may describe *all* possible ways topological consensus networks can interact as a series of gluing surgeries between networks. Given a single network, how can it change in time given rounds of consensus which have the possibility of breaches in trust? We formulate this as the interaction of a single network history with itself as the series of cobordisms [of the same class]:

Figure 22. 2-point function for topological consensus networks. Given an initial and final state of a single consensus network, this combinatoric series represents all possible ways the network can split amongst itself into sub-networks, and then recombine into a single network. Furthermore, it demonstrates all possible trust breaching events following rounds of consensus.

More uncertainty in the trust of the network leads to increased splittings and, consequently, a higher genus. Accordingly we denote *branches* — independent networks evolutions from the same source — of interactions as more trustworthy given a minimal genus. Moreover, we can also consider how two independent networks [and hence their histories] can interact given consensus rounds. We characterize this by the diagrammatic series:

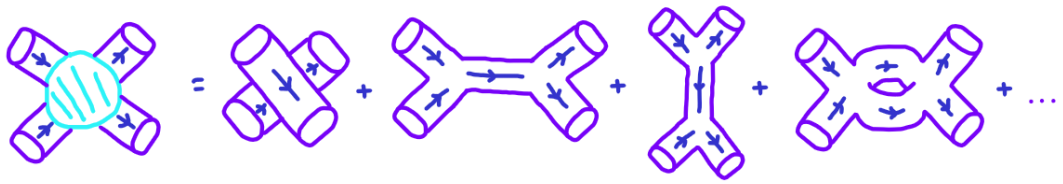


Figure 23. 4-point function for topological consensus networks. Given an initial and final state of a two consensus networks, this combinatoric series represents all possible ways the network can split amongst itself into sub-networks, and then recombine into two separate networks.

It is noted that the third term in the series represents part of networks *shedding* sizes. All diagrams are drawn with the same thickness as the diagram sizes of the initial and final networks are implied by the direction of flow. As another example, we can consider restricting our interactions to a cobordism class Ω that has three initial disconnected pieces, and one final. This would represent the way in which three separate networks interact to form a single network. Once again we represent it as the combinatoric series of surgeries:

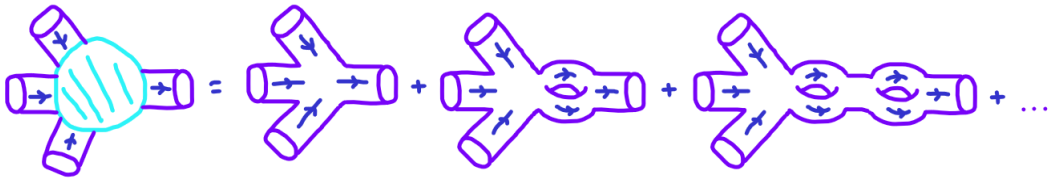


Figure 24. All possible evolutions of a system of three topological consensus networks ultimately combining into a single network after events of bifurcation for each consensus round.

Thus the prescription for how networks interact and what information is required specifying unique network evolutions is given by the following. Given an initial amount of independent topological consensus networks, one specifies a cobordism class Ω which has disconnected boundary pieces $\mathcal{B}$. For a class with N initial networks [open boundary subsets] and M final networks after interactions, we say we have an interaction of *type* $N \to M$ [the two and four point functions are denoted as $1 \to 1$, $2 \to 2$, respectively, while the interaction in Fig. 24 is written as $3 \to 1$]. After specifying the interaction type, we write all possible network evolutions of given boundary conditions of the disconnected pieces of $\mathcal{B}$ as a combinatoric series of cobordisms. We can compute the probabilities off all possible evolutions, and can select which to use to uniquely specify a network evolution. Now that we have such a prescription, we can look how it applies to **network autonomy**. For a given network $\mathcal{G}$ [and corresponding history $\mathcal{H}$], what possible histories does the network *want* to combine with? We would imagine one in a space of histories that is most trustworthy. We can characterize the *distrust* of a network history by its *genus* g — the maximum number of non-self-intersecting closed curves that one can draw on the surface without disconnecting it — given the topologies of its different branches. Thus a network searches for another network with minimal genus and checks if their type is compatible. For a network $\mathcal{G}$ of type $N \to M$, another network $\mathcal{G}'$ is *compatible* with it if the amount of incoming pieces of $\mathcal{G}'$ is the same amount of outgoing pieces of $\mathcal{G}$ [meaning the type

for $\mathcal{G}'$ must be type of $M \to P$ for some arbitrary amount of outgoing pieces P]. This means that locally networks can specify a type and level of trust to autonomously interact amongst other networks without the need for external input. This interaction is captured by a diagramatic series of cobordisms in which interaction vertices are glued together. For a system of many independent networks, later we will see that the specification of a type will *additionally* require other information such as the topological invariants of the network histories. Next, we elaborate on the topological formulation of trust in terms of topological invariants in a continuous system.

2.2.3 Topological invariants of histories

In the first section we described distrust as being characterized by the number of bifurcations formed after compliance checks at the end of consensus events. We attributed what is meant by distrust to the genus of a network history. Although our system of nodes and transactions are inherently discrete, we opted for a *continuous* description such that we can define smooth network histories and make use of continuous topological invariants. Furthermore, we didn't explicitly discuss how one computes trust given arbitrary network history. We address these topics in this section.

We motivated the use of equivalence classes of compact manifolds [cobordism classes Ω] to describe the set of all topological consensus networks which can be deformed [evolved] into one another. We can infer more topological information from the network histories by studying other equivalence classes such as that of *forms* and *cycles*. For this we must look at *cohomology* and *homology* groups [11], respectively, over some history $\mathcal{H}$. We begin by introducing *de Rham* cohomology, followed by *simplicial* homology.

First, we motivate cohomology groups by looking at *vector spaces*. To look at a vector space over a history $\mathcal{H}$ is to look at a *decomposition with some sets of rules*. A vector space V over $\mathcal{H}$ is constructed such that it is a decomposition into subspaces $\{V^a\}$ via summation:

$$V(\mathcal{H}) = \bigoplus_a V^a(\mathcal{H}). \tag{2.38}$$

Here $\bigoplus$ refers to summing the over the subsets $V = \sum_a V^a$ under which its elements [vectors] adhere to a uniqueness condition $\vec{v}^b \cap \sum_{a\neq b} \vec{v}^a = \{0\}$, $\forall\ \vec{v}^a \in V^a$. The uniqueness condition makes the summation what is called a called a *direct sum*. This is related to constructing representations in quantum theory as the sum of irreducible representations[†]. Over this vector space one has an orthonormal basis for V written as $\{\vec{e}_a\}$ from which can decompose a vector $\vec{v} \in V$ as:

[†]Consider two irreducible matrix representations [such as Pauli matrices for $SU(2)$] $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}$. To come up with a matrix that acts on an n^2-dimensional vector representation, we consider the direct sum of the irreducible representations as:

$$\mathbf{A} \oplus \mathbf{B} = \begin{bmatrix} \mathbf{A} & 0 \\ 0 & \mathbf{B} \end{bmatrix}.$$

$$\vec{v} = \sum_a (\vec{v} \cdot \vec{e_a})\vec{e}_a, \tag{2.39}$$

for some real coefficients c^a. Typically the convention is to write $(\vec{v} \cdot \vec{e}_a) \equiv v^a$, where v^a are the *vector components* of $\vec{v}$, and so with some shorthand notation — and dropping the vector hats — we instead have: $v = v^a e_a$ [where the sum is implied over a]. Where exactly do these vectors live? Consider a point $p \in \mathcal{H}$ such that we consider the tangent space at that point, $T_p\mathcal{H}$. The tangent space corresponds to a vector space in which all vector tails terminate at p, and all vectors are tangent to $\mathcal{H}$ at p. The basis of such is given by partial derivatives $\{\partial_a\} = \{\partial/\partial y^a\}$ for coordinates $y^a = (t, n, \tau)$, which correspond to elements in the time, nodal, and transaction dimensions $(\mathbb{R}, \mathcal{N}, \mathcal{T})$, respectively. In this we represent a vector as $v = v^a\partial_a$. *The logic behind the derivative basis is that it is the same as taking directional derivatives of functions on curves which give the tangential rate of change.* Consider the derivative of a function $f \in C^\infty(\mathcal{H})$ at a point $p \in \mathcal{K}$ in some curve $\mathcal{K} \subset \mathcal{H}$. Being that $\mathcal{H}$ is higher dimensional, the derivative of a function takes into account all the directions in which the function changes [as it is acted on by the flow which the tangential vector fields induce]. To select a *unit* direction v, we act the vector on the function in the form of a scalar product between v and the gradient of f:

$$v(f) = v \cdot \nabla f = v^a\partial_a f = -v^t\frac{\partial f}{\partial t} + v^n\frac{\partial f}{\partial n} + v^\tau\frac{\partial f}{\partial \tau}. \tag{2.40}$$

Notice the minus sign in front of the first term as we are using Lorentzian manifolds to describe network histories. Here v is in fact an operator which acts on functions f in the form $v : C^\infty(\mathcal{H}) \longrightarrow \mathbb{R}$. We can write the components v as $v = v^a\partial_a$, which acts on functions f as $v(f) = (v^a\partial_a)f$. Thus we see in this case then the basis of which v is expressed in is the basis of partial derivatives. More generally, instead of vectors we can instead have *tensors* which can be interpreted as higher rank vectors [such as matrices] which transform under the Jacobian. In this case instead of vector coefficients v^a, we instead have tensor coefficients $v^{a\ldots b}$. If the coefficients are symmetric such that $v^{a\ldots b} = v^{b\ldots a}$, then we may represent the tensor in the tangent basis:

$$v = v^{a\ldots b}\,\partial_a \otimes \cdots \otimes \partial_b, \tag{2.41}$$

where $\otimes$ is known as the *tensor product* which roughly speaking is a generalization of the outer product of vectors which allows you to combine the information of an arbitrary amount of tensors. If the coefficients of the tensor are antisymmetric such that $v^{a\ldots b} = -v^{b\ldots a}$ then we are instead working with a basis of *forms*. Forms ω are linear functionals which intake vectors to produce scalars such as $\omega(v) \in \mathbb{R}$. While vector spaces at a point p — denoted as V_p — are defined over the tangent space $T_p\mathcal{H}$, the space of forms Ω_p at a point p are defined over cotangent spaces $T_p^*\mathcal{H}$. We denote the components of an antisymmetric tensor [a form] with lowercase indices as $\omega_{a\ldots b}$. We say that the tangent and cotangent spaces are **dual** to each other being that there is a relation between antisymmetric and symmetric tensors through a bilinear symmetric tensor known as a *metric* $g : T_p\mathcal{H} \times T_p\mathcal{H} \longrightarrow \mathbb{R}$. The relation is given as:

$$(g \circ \cdots \circ g)(\omega) = v, \tag{2.42}$$

where the number of compositions of the metric g is given by the amount of indices [the *rank*] of ω. While the basis of the tangent space is given by partial derivatives, the basis of the cotangent space will comprise of measures dy^a. What do these mean exactly? Recall from vector calculus that a change in variable $z^a = \gamma y^a$ — for some $\gamma \in \mathbb{C}$ — results in measures for integrals as $dz^a = \gamma dy^a$. These are in fact *exterior derivatives* d [generalizations to operations such as gradients, divergence, and curl, depending on the dimension of the space] which act on forms [in our case coordinates]. To put explicitly, 0-form is a function while a 1-form is a function multiplying a basis measure such as $\omega = \omega_a dy^a$. A two form is a contraction of indices with two basis forms $\omega = \omega_{ab} dy^a \wedge dy^b$, where the $\wedge$ operation is known as a *wedge* product and is the anti-symmetrization of the tensor product $\otimes$. More generally, an n-form ω lives in the space of n-forms, written as $\Omega^n(\mathcal{H})$. Much like the decomposition of a vector space, we decompose the space of all forms over $\mathcal{H}$ as $\Omega(\mathcal{H}) = \bigoplus_a \Omega^a(\mathcal{H})$. An element $\omega \in \Omega$ can be expressed in the wedge product basis of 1-forms [measures]:

$$\omega = \omega_{a\ldots b}\, dx^a \wedge \cdots \wedge dx^b. \tag{2.43}$$

The relationship between the basis of forms and vectors is given as: $dy^a(\partial_b) = \delta^a_b$. While symmetric vectors give us information of flow, forms give us information flow through pieces of space such as flux. The dual spaces — in the form tangent and cotangent spaces — are related via a musical isomorphism and so we can analogously extract information from either representation. Finally to discuss cohomology groups, we review some distinct sets of forms. If a form ω vanishes under the action of an exterior derivative d, then we call it *closed* such that $d\omega = 0$. If a form ω can be written as an exterior derivative of a lower rank form β [such as $\omega = d\beta$], then we say it is *exact*. Let $C^n(\mathcal{H}) = \{\omega_n : d\omega_n = 0\}$ and $E^n(\mathcal{H}) = \{\nu_n : \nu_n = d\alpha_{n-1}\}$ be the set of closed and exact n-forms over $\mathcal{H}$, respectively. We can construct the n-th de Rham cohomology group over $\mathcal{H}$ as the quotient set:

$$H^n(\mathcal{H}) = C^n(\mathcal{H})/E^n(\mathcal{H}). \tag{2.44}$$

Here the elements of H^n are equivalence classes of closed n-forms on $\mathcal{H}$, where the forms of the partitions are considered equivalent if they differ by an exact form:

$$\omega_n \sim \omega_n + d\alpha_{n-1}. \tag{2.45}$$

The cohomology group actually tells us quite a bit about the topology of $\mathcal{H}$, but this might seem too intuitive to think about this in terms of forms. We thus turn our attention to *simplicial homology* groups. Consider n-dimensional submanifolds $\{\mathcal{H}_i\}$ of $\mathcal{H}$, each labelled by an index i. We can consider what is known as an *n-chain* a_n, which is the sum over the submanifolds of $\mathcal{H}$:

$$a_n = \sum_i c_i \mathcal{H}_i, \tag{2.46}$$

where $c_i \in \mathbb{C}$ are coefficients. An *n-cycle* is an n-chain that does not have a boundary such that $\partial a_n = 0$. From this we will classify again a particular set of distinct cycles. Let $C_p(\mathcal{H}) = \{a_p : \partial a_p = 0\}$, and $B_p(\mathcal{H}) = \{b_p : b_p = \partial b_{p+1}\}$ be the set of n-cycles and n-boundaries [n-chains which are boundaries to manifolds] of $\mathcal{H}$, respectively. The n-th simplicial homology group over $\mathcal{H}$ can thus be written as the quotient set:

$$H_n(\mathcal{H}) = C_n(\mathcal{H})/B_n(\mathcal{H}). \tag{2.47}$$

Here the elements of H_n equivalence classes of n-cycles of $\mathcal{H}$, where two elements of a partition are equivalent if they differ by a boundary:

$$a_n \sim a_n + \partial c_{n+1}. \tag{2.48}$$

While the equivalence of forms has no immediate physical interpretation, we can have some sense of intuition for equivalences in homology groups. Consider the homology of a torus as [11]:

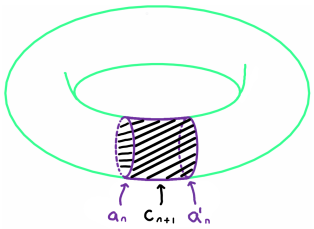


Figure 25. Visualization of the homology of a torus. Here a_n and a'_n are n-cycles of the torus [in our case $n = 2$], while c_{n+1} is a submanifold of the torus. The cycles a_n and a'_n are equivalent up to the boundary of the submanifold which separates them, given by ∂c_{n+1}. Thus we say a_n is equivalent to $a'_n \equiv a_n + \partial c_{n+1}$ as in E.q. (2.48).

What does this have to do with topology? Well first off we can construct the topological invariants based on these groups, which are quantities that are preserved under continuous mappings or homeomorphisms of the space. For example, the dimension of the cohomology groups are the *Betti numbers* given by $b_n = \dim H^n$, which tell us the number of linearly independent harmonic[†] n-forms on $\mathcal{H}$. Additionally, this describes the amount of irreducible n-cycles of $\mathcal{H}$. The connection between the homology and cohomology groups of $\mathcal{H}$ is given by the *Poincaré duality*, which is an isomorphism between the groups:

$$H^n(\mathcal{H}) \cong H_{m-n}(\mathcal{H}), \tag{2.49}$$

which holds if $\mathcal{H}$ is a compact manifold for $m = \dim \mathcal{H}$, and $n \in \mathbb{Z}_+$. Although it might

[†]Harmonic forms vanish under the action of the Laplacian Δ as $\Delta\omega = 0$.

not seem too informative, the cohomology group tells us what forms[‡] can exist on $\mathcal{H}$, and the forms correspond to field operators in QFT. These field operators excite vacua to give rise to particles, and so we say the topology of the space $\mathcal{H}$ tells us exactly what kind of particles can exist on it. An example is the unit 2-sphere S^2 which has the Betti numbers $b_0 = 1, b_1 = 0, b_2 = 1$. Here $b_1 = 0$ tells us that S^2 does not admit a global 1-form or dual vector field, which is a manifestation of the *hairy ball theorem*. This is a direct result of the topology as if we punctured the unit sphere and deformed it to instead be a 2-torus T^2, b_1 would no longer vanish. In essence, the cohomology and homology groups tell us what forms and submanifolds of $\mathcal{H}$ are allowed to exist and in turn gives us information about it's topology.

Now, back to **trust**. Why do we associate trust with genus? We can think of genus as representing the number of times a network has split and recombined which has shakier grounds for its trust as opposed to a network that has never split. We say that the trust has to do with the genus g of the network history which is related to it's Euler characteristic χ via: $\chi = 2 - 2g$. Formally, we can write the Euler characteristic of a network history $\mathcal{H}$ via an alternating sum of its Betti numbers:

$$\chi(\mathcal{H}) = \sum_{a=0}^{\dim \mathcal{H}} (-1)^a b_a(\mathcal{H}) = \sum_{a=0}^{\dim \mathcal{H}} (-1)^a \dim H^a(\mathcal{H}). \tag{2.50}$$

Thereafter, we say the amount of *distrust* [how dishonest a network is perceived as; the additive inverse of trust] of a consensus network history $\mathcal{H}$ is given by its genus:

$$r(\mathcal{H}) \equiv g(\mathcal{H}) = 1 - \frac{1}{2}\chi(\mathcal{H}). \tag{2.51}$$

A trusthworthy network is one which *minimizes* this quantity along the different *branches* [distinct network bifurcations of a given history]. What does distrust look like for combined network histories which have interacted combinatorically? Consider indexing a set of network histories with indices (i, j) and we select two such cobordisms $(\mathcal{H}_i, \mathcal{H}_j)$ which are compatible as they are of the same type. We glue them along open subsets $(\partial\mathcal{H}_i, \partial\mathcal{H}_j)$ with the use of diffeomorphisms $\phi = \{\phi_{ij} : \partial\mathcal{H}_i \to \partial\mathcal{H}_j\}$ mapping between them which we write as an identification $\partial\mathcal{H}_i \sim \partial\mathcal{H}_j$. Recall that their union is defined in terms of a disjoint union modulo an equivalence relation on these open subsets:

$$\mathcal{H}_{ij} = \mathcal{H}_i \cup_\phi \mathcal{H}_j = (\mathcal{H}_i \sqcup \mathcal{H}_j) / \sim . \tag{2.52}$$

How would one compute the Euler characteristic for such a space? Had we no identifications of boundary [i.e. the case of $\mathcal{H}_{ij} = \mathcal{H}_i \sqcup \mathcal{H}_j$] then by the inclusion-exclusion principle of Euler characteristics, we would simply have $\chi(\mathcal{H}_{ij}) = \chi(\mathcal{H}_i) + \chi(\mathcal{H}_j)$. However we have a slightly non-trivial case of identification which does not make us allowed to use that. For this we will make use of *Mayer-Vietoris sequences* [12] which provides a way to compute the cohomology of the union of two open sets [in our case is the disconnected

[‡]For each n-form we have a corresponding rank n tensor field given by the *musical isomorphism* which maps between the cotangent and tangent spaces of $\mathcal{H}$.

boundaries of the network histories]. We can link n-th cohomology groups of histories using this as the exact sequence [a sequence of maps such that the image of one map equals the kernel of the next]:

$$\cdots \longrightarrow H^n(\mathcal{H}_{ij}) \longrightarrow H^n(\mathcal{H}_i) \oplus H^n(\mathcal{H}_j) \longrightarrow H^n(\partial\mathcal{H}_i) \longrightarrow H^{n+1}(\mathcal{H}_{ij}) \longrightarrow \ldots. \tag{2.53}$$

From this sequence one can deduce based on arguments of dimensions of the cohomology groups that the Euler characteristics are related by:

$$\chi(\mathcal{H}_{ij}) = \chi(\mathcal{H}_i) + \chi(\mathcal{H}_j) - \chi(\partial\mathcal{H}_i). \tag{2.54}$$

Note since we have the identification $\partial\mathcal{H}_i \sim \partial\mathcal{H}_j$, then equivalently the last term above can be instead written as $-\chi(\partial\mathcal{H}_j)$. Thus, the **distrust of combined network histories** — for two network histories $(\mathcal{H}_i, \mathcal{H}_j)$ is given by:

$$r(\mathcal{H}_{ij}) = 1 - \frac{1}{2}\bigg(\chi(\mathcal{H}_i) + \chi(\mathcal{H}_j) - \chi(\partial\mathcal{H}_i)\bigg). \tag{2.55}$$

It should be noted that the network history represents the understanding of trust for a given *perspective*, which can be the viewpoint of a single node, an entire network, or a client. For an i-th node of a topological network, we denote its distrust in the j-th node as we r_{ij}. We say the total distrust of node i with all other j is given by the trace $r_i = \mathrm{tr}_j(r_{ij})$, and [for the case of a non-topological network $\mathcal{N}$] with this we can compute the total amount of distrust in a network:

$$r(\mathcal{N}) = \frac{1}{|\mathcal{N}|}\sum_{i=1}^{|\mathcal{N}|} r_i = \frac{1}{|\mathcal{N}|}\sum_{i=1}^{|\mathcal{N}|}\sum_{j=1}^{|\mathcal{N}|} r_{ij}. \tag{2.56}$$

In our case of a topological network $\mathcal{G}$, we must additionally include the information of the transactions to adhere to the 2D structure of the topological network, which we will label with indices k. We denote the distrust node i has in other nodes regarding transaction k is given by: $r_{ik} = \mathrm{tr}_j(t_{ijk})$. Thus the total distrust of a topological consensus network among its elements is:

$$r(\mathcal{G}) = \frac{1}{|\mathcal{G}|}\sum_{i=1}^{|\mathcal{N}|}\sum_{k=1}^{|\mathcal{T}|} r_{ik} = \frac{1}{|\mathcal{G}|}\sum_{i=1}^{|\mathcal{N}|}\sum_{j=1}^{|\mathcal{N}|}\sum_{k=1}^{|\mathcal{T}|} r_{ijk}. \tag{2.57}$$

One can consider distrust that evolves in time which we represent via a discrete index t and so our higher dimensional matrix of distrust becomes r^t_{ijk}. The time index labels the different consensus [and so compliance check] events and so we say the distrust in a network at a discrete time t is:

$$r(\mathcal{G}; t) = \sum_{t'=0}^{t} \frac{1}{|\mathcal{G}^{t'}|}\sum_{ijk} r^{t'}_{ijk} \tag{2.58}$$

Here $\mathcal{G}^t$ is the consensus network at discrete time t, and t' is a temporary index to sum over all time until t. We can convert this discrete sum into an integral by considering the continuous underlying space of the simplicial complex that describes the consensus network and instead recover:

$$r[\mathcal{G}](t) = \int_0^t dt' \, \frac{1}{|\mathcal{G}(t')|!} \int_{\mathcal{G}} \mathcal{D}n\mathcal{D}n'\mathcal{D}\tau \; r(t; n, n', \tau). \tag{2.59}$$

Here our discrete indices (i, j, k, t) have been replaced with continuous variables (n, n', τ, t), where n' represents a node other than n. The reason for switching to an integral is to describe the complete continuous evolution of the system as opposed to discrete time steps. The equation for $r(\mathcal{G})$ gives us the distrust in a topological consensus network at a given instance in time. As it evolves and bifurcates — forming a network history $\mathcal{H}$ — then the distrust of its history is: $r(\mathcal{H}) = 1 - \frac{1}{2}\chi(\mathcal{H})$. One could imagine that the connection between the trust is that the information of its topology should agree with the individual trust of nodes in the network as it evolves in time. By this accord, we postulate for a time-dependent topological consensus network $\mathcal{G}(t)$ [which traces out $\mathcal{H}$], the total amount of distrust in a history is given by the relation:

$$r(\mathcal{H}) = 1 - \frac{1}{2}\chi(\mathcal{H}) \stackrel{?}{=} \int dt \; r[\mathcal{G}](t). \tag{2.60}$$

This is a question of the relation between the different graining of information at different scales. Here we question whether or not the information at the instantaneous fine scale is directly related to the global coarse scale of topological information. Whether or not it is true is independent from the fact that these separate pieces of information gives us a bigger picture for what is going on. Furthermore, given a network which has bifurcated and reassembled based on conditions on distrust, it [or rather its continuous underlying structure from its simplicial complex] traces out a continuous network history $\mathcal{H}$. Recall that we define the distrust of $\mathcal{H}$ in another history as its genus through its Euler characteristic as: $r(\mathcal{H}) = 1 - \frac{1}{2}\chi(\mathcal{H})$. From this, if we consider another network history $\mathcal{Q}$, we can analogously as before define a binary decision outcome on whether or not the two histories should combine [or cooperate]:

$$f_{\mathcal{H}}(\mathcal{Q}) = \begin{cases} 1, \; r(\mathcal{Q}) \leq \delta_{\mathcal{H}} \\ 0, \; r(\mathcal{Q}) > \delta_{\mathcal{H}}, \end{cases} \tag{2.61}$$

for a network [agreed] history security parameter $\delta_{\mathcal{H}}$. The parameter would be agreed upon by the latest topological network $\mathcal{G}$ based on their own history. Should these networks combine such that $(\mathcal{H}, \mathcal{Q}) \longrightarrow \mathcal{H} \cup_\phi \mathcal{Q}$, their updated distrust is given instead by a linear combination of Euler characteristics:

$$r(\mathcal{H} \cup_\phi \mathcal{Q}) = 1 - \frac{1}{2}\Big(\chi(\mathcal{H}) + \chi(\mathcal{Q}) - \chi(\partial\mathcal{H})\Big), \tag{2.62}$$

where $\partial\mathcal{H}$ represents the boundary of the network history $\mathcal{H}$, or rather the collection

of all initial and final topological consensus networks.

What about an external system of clients? If we have some internal system of independent topological consensus networks [each evolving to trace out their respective histories], how can we utilize this to be used by some external client wanting to securely verify a given transaction? Well first me must consider that the notion of trust is **not** a local concept. The network's local conception of trust given as a topological network history maybe not coincide with the client's external independent conception of trust among nodes in the network [much like how nodes have different levels of trust within one another]. Thus to make an autonomous system for clients to use, we must consider the intersection of trust among networks and clients. As an example, we can consider the scenario in which the network and client completely disagree on which subsets of networks are trustworthy:

Figure 26. Null network history trust intersection. Sub-networks which are trusted are coloured magenta while un-trusted teal. The left prong represents trust based on the network's perspective while the right prong represents client's perspective of trust. The trust intersection [represented by the $\cap_T$ operator] is thus empty as the network and client completely disagree on which topological consensus sub-networks are trustworthy.

Alternatively, we can have a non-trivial intersection in which a subset of a trusted network is mutually selected from both perspectives:

Figure 27. Real network history trust intersection. In this case on the left we have the network splitting into three sub-networks, two of which are un-trustworthy to different degrees. On the client's side, the rightmost [network] un-trusted network is in fact trusted. Thus the intersection is simply the trusted network as viewed by the nodes' perspective.

$$\mathfrak{ABCDEFGHIJKLMNOPQRSTUVWXYZ} \tag{4.1}$$

In this section we go over the details of how algorithms mentioned in the paper work which includes the motivation for how the algorithm was made, and how it is executed. All code will be linked from the **BTQ Quantum GitHub** repository but will not be appear here explicitly.

We begin by looking at the *network partitioning protocol* [algorithm 2] and shed light on its inner workings. Recall the point is to take a distribution of networks $\{\mathcal{N}\}$ — along with the trust matrix r_{ij} and security parameters $\{\delta_i\}$ — and from those pieces of information partition the system into a more optimal distribution of networks.

This is a deprecated section to propose the logic of switching from a discrete system to a continuous system, similar to how is done in statistical mechanics. However for the purposes of studying topology at different scales, the use of simplicial complexes [and their underlying continuous structures] was deemed more appropriate.

First, we must justify our use of a continuum limit on a discrete system. Known as a *scaling limit* in mathematics, the continuum limit[†] of a lattice spacing [the distance between lattice sites] δ is the limit in which we take $\delta \to 0$. A popular application of this occurs when discussing discrete random processes such as Brownian motion which can be

[†]One could have for example a continuous quantum field theory as approximated by a lattice model in the limit where the lattice spacing vanishes. Such a process corresponds to finding a second order phase transition of the model.

approximated as a continuous process for late times. For our case of a network history, then before the continuous limit we have a discrete set $\mathcal{Y}$ combined with a continuous set $\mathbb{R}$ as $\mathcal{M} = \mathbb{R} \times \mathcal{Y}$. If the spacing between transactions is given by $\delta_{\mathcal{T}}$ and the spacing between nodes is given by $\delta_{\mathcal{N}}$, then we say the continuum limit of this system is given by: $\delta_{\mathcal{T}}, \delta_{\mathcal{N}} \to 0$. Can we be more precise about this? Consider the *Whitney embedding theorem* [13]:

i) Any smooth real m-dimensional manifold can be smoothly embedded in $\mathbb{R}^{2m}$ [as the continuum limit of $\mathbb{Z}^{2m}$] for $m > 0$.

ii) Any continuous function from an n-dimensional manifold to an m-dimensional manifold may be approximated by a smooth embedding provided $m > 2n$.

For our purposes we will consider an embedding of our smooth topological consensus network in a higher dimensional discrete space for which we have taken the continuous limit. Now, what does it look like when we take such a limit? To answer this we will take motivation from *statistical mechanics*. Consider a discrete system of particles [an ensemble] of dimension d, linear size L [the length of a side of the system], and correlation length η [approximately the size of particles]. We say that the number of independent parts in this macroscopic system is:

$$N = \left(\frac{L}{\eta}\right)^d. \tag{B.1}$$

If we consider the system as having many possible states, then the total amount of possible states is a combinatoric result: 2^N. The information of the dynamics of the system is succinctly captured by the *partition function* Z which sums over all possible states of the system. If we denote the different states of the system as σ [which can be interpreted for example as energy states E_σ] then we define the partition function as:

$$Z = \sum_\sigma e^{-E_\sigma/kT}, \tag{B.2}$$

where E_σ is the energy of a given state, k is the Boltzmann constant, and T is the temperature of the system. From Z we can compute many macroscopic quantities by differentiating it, and furthermore is used as a normalization for computing observables as expectation values. Now what if we wanted to take the continuum limit, how would this change? If we consider the volume of the phase space per particle as V [hence the total volume of N particles scales as V^N], we must take into account overcounting identical particles which comes with a factor of $N!$. Thus we say for the large N limit, the total volume is given by $V^N/N! \approx (V/N)^N$. The volume of the phase space correspond exactly to counting all states of the system: $V^N = \sum_\sigma$. If we want to compute this for all particles in the system, then we must sum over all degrees of freedom. Being that we are working in the continuum limit [$N \gg 1$], this corresponds to integrating over all positions and momenta of the system as:

$$\sum_{\sigma} = \frac{1}{N!}\prod_{a=1}^{N}\int \frac{d\vec{q}_a\, d\vec{p}_a}{(2\pi\hbar)^d}. \tag{B.3}$$

Here then $\vec{q}_a$ corresponds to the d-dimensional position vector for the a-th particle, $\vec{p}_a$ corresponds to the momentum for the a-th particle, and the denominator is a normalization that comes from the Heisenberg uncertainty principle[†]. The measure $d\vec{q}_a$ can be equivalently expressed as $d^d q_a$. Our partition thus becomes in the continuum limit:

$$Z = \frac{1}{N!}\prod_{a=1}^{N}\int \frac{d\vec{q}_a\, d\vec{p}_a}{(2\pi\hbar)^d} \exp\left(-\frac{1}{kT}\sum_{a=1}^{N} H[\vec{q}_a, \vec{p}_a]\right). \tag{B.4}$$

Here H is the Hamiltonian [roughly speaking the energy functional] of the system which contains the information of the energy states of the ensemble and hence all of its dynamics [the Hamiltonian generates time translations as given by Noether's theorem [14]]. Thus we see in essence the *continuum limit* amounts to a different way of summing the information of the system; we sum over infinitesimally separated pieces of information instead of summing discretely separated points. For our purposes then integrating over the phase space of the topological consensus networks amounts to:

$$\frac{1}{|\mathcal{Y}|!}\prod_{a}^{|\mathcal{N}|}\prod_{b}^{|\mathcal{T}|}\int_{\mathcal{Y}} dn_a d\tau_b. \tag{B.5}$$

Here $n_a \in \mathcal{N}$ is a node, and $\tau_b \in \mathcal{T}$ is an associated transaction. In the spirit of path integral measures, we shall define them as such:

$$\prod_{a}^{|\mathcal{N}|} dn_a, \prod_{b}^{|\mathcal{T}|} d\tau_b \equiv \mathcal{D}n, \mathcal{D}\tau. \tag{B.6}$$

Furthermore, since we are only interested over the topological consensus network $\mathcal{G}$ embedded in $\mathcal{Y}$, we will instead integrate over $\mathcal{Y}$ restricted over $\mathcal{G}$ [implicitly we assume whatever function we integrate over has compact support over $\mathcal{G}$]. Thus the phase space of a topological consensus network is given by:

$$V_{\mathcal{G}} = \frac{1}{|\mathcal{G}|!}\int_{\mathcal{G}} \mathcal{D}n\mathcal{D}\tau \tag{B.7}$$

If we wanted to integrate over the complete information of a network history, we need only include a temporal measure of the form dt and thus integrate over $\mathcal{M}$ instead of $\mathcal{Y}$ [and correspondingly restrict this integral over the domain of compact support which is the network history $\mathcal{H}$]. Now that we have a prescription for taking the continuum limit

[†]The product measure over the degrees of freedom of the particles are precisely path integral measures in quantum field theory [related when rotating to real time via a Wick rotation]:

$$\prod_{a=1}^{N}\frac{d\vec{q}_a}{(2\pi\hbar)^{d/2}}, \prod_{a=1}^{N}\frac{d\vec{p}_a}{(2\pi\hbar)^{d/2}} \equiv \mathcal{D}\vec{q}, \mathcal{D}\vec{p}.$$

of our originally discrete system, we should now discuss what is meant by *trust* and how we can compute it with phase space integrals. To evaluate the trust of a network from the perspective of another we require a classification of histories up to diffeomorphisms [isomorphisms of smooth manifolds]. We saw that breaches in trust result in bifurcation and thus the creation of non-trivial genus. In this sense we say that the **distrust** of a topological consensus network history is precisely given by its genus. How do we characterize a genus of a manifold other than naively counting its holes? Incomes the *Euler characteristic* χ. For us to understand this we must brush up on a few more topological concepts: *homology and cohomology.*

References

[1] D. Singh, G. Muraleedharan, B. Fu, C.-M. Cheng, N.R. Newton, P.P. Rohde et al., *Proof-of-work consensus by quantum sampling*, 2305.19865.

[2] F. Chazal and B. Michel, *An introduction to topological data analysis: Fundamental and practical aspects for data scientists*, *Frontiers in Artificial Intelligence* **4** (2021) .

[3] Z. Yang and F. Cohen, *Image registration and object recognition using affine invariants and convex hulls*, *Image Processing, IEEE Transactions on* **8** (1999) 934 .

[4] R. Brandenberger, *Topological defects and structure formation*, *International Journal of Modern Physics A* **09** (1994) 2117.

[5] R. Brown, P.J. Higgins and R. Sivera, *Nonabelian Algebraic Topology*, EMS Press (Aug., 2011), 10.4171/083.

[6] A. Hatcher, *Algebraic Topology*, Cambridge University Press, Cambridge, England (Dec., 2001).

[7] J. Baez and J.P. Muniain, *Gauge Fields, Knots and Gravity*, World Scientific (Oct., 1994), 10.1142/2324.

[8] J.C. Baez and J. Vicary, *Wormholes and Entanglement*, *Classical and Quantum Gravity* **31** (2014) 214007.

[9] C.T.C. Wall, *Determination of the cobordism ring*, *The Annals of Mathematics* **72** (1960) 292.

[10] S. Smale, *On the structure of manifolds*, *American Journal of Mathematics* **84** (1962) 387.

[11] P. Candelas, *Lectures on Complex Manifolds*, Springer-Verlag, 1st ed. (1987).

[12] A.I. Generalov, *Algebraic mayer–vietoris sequence*, *Journal of Mathematical Sciences* **264** (2022) 39–43.

[13] H. Whitney, *Differentiable manifolds*, *The Annals of Mathematics* **37** (1936) 645.

[14] E. Noether, *Invariante variations probleme, Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* **1918** (1918) 235.