Scaling Up Structural Clustering to Large Probabilistic Graphs Using Lyapunov Central Limit Theorem

Joseph Howie
University of Victoria
Victoria, Canada
joehowie@uvic.ca

Venkatesh Srinivasan
University of Victoria
Victoria, Canada
srinivas@uvic.ca

Alex Thomo
University of Victoria
Victoria, Canada
thomo@uvic.ca

ABSTRACT

Structural clustering is one of the most widely used graph clustering frameworks. In this paper, we focus on structural clustering of probabilistic graphs, which comes with significant computational challenges and has, so far, resisted efficient solutions that are able to scale to large graphs, e.g. the state-of-art can only handle graphs with a few million edges. We address the main bottleneck step of probabilistic structural clustering, computing the structural similarity of vertices based on their Jaccard similarity over the set of possible worlds of a given probabilistic graph. The state-of-art used Dynamic Programming, a quadratic run-time algorithm, that does not scale to pairs of vertices of high degree. In this paper we present a novel approach based on Lyapunov Central Limit Theorem. By using a carefully chosen set of random variables we are able to cast the computation of structural similarity to computing a one-tailed area under the Normal Distribution. Our approach has linear runtime as opposed to quadratic, and as such, it scales to much larger inputs. Extensive experiments show that our approach can handle massive graphs at web-scale which the state-of-art cannot.

PVLDB Reference Format:
Joseph Howie, Venkatesh Srinivasan, and Alex Thomo. Scaling Up Structural Clustering to Large Probabilistic Graphs Using Lyapunov Central Limit Theorem. PVLDB, 16(11): 3165 - 3177, 2023.
doi:10.14778/3611479.3611516

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/JoetheManHowie/nuscan.

1 INTRODUCTION

Probabilistic graphs are graphs in which each edge has a probability of existence. The uncertainty associated with this data structure allows for the modelling of many natural phenomena which have probabilistic interactions. The literature is rich with analyses of foundational data mining operations cast into the realm of the probabilistic graphs [1, 2, 15, 16, 20, 22–24, 27, 34, 47, 48]. For instance, in social networks the influence users have over one another represents a probability of information passing between users of the network [30, 39]. With online dating networks, probabilistic graphs can model the likelihood that a user will visit another user's profile and whether they will send a message to said user [32, 41]. In the study of protein-protein interactions, experimental procedures determine the connections formed with a measurement uncertainty which is interpreted as the edge probability [25, 36].

For large graphs, a popular data mining operation is clustering, which groups similar vertices in the same cluster and separates dissimilar vertices into different clusters. A widely used approach for clustering deterministic graphs is the Structural Clustering Algorithm for Networks (SCAN) [42]. What distinguishes this algorithm from other clustering approaches is that it allows the clusters to overlap and furthermore introduces vertex classification. Namely, the algorithm defines three types of vertices: core, hubs, and outliers; and uses a metric to determine the type of each vertex in the network. Clusters are formed by grouping core vertices together based on maximal connectivity. After the clusters are formed, vertices that do not belong to any cluster are either hubs or outliers based on whether the vertex has edges connecting to multiple or just one cluster respectively. With these labels for vertices, great use has been made of the SCAN method on problems such as: community detection on population networks, fraud detection in financial networks, and on protein-protein interactions in biological networks [8, 33]. The SCAN method has a strong foundation in the literature, with many works expanding upon the original framework by modifying the similarity metric, and implementing parallel processes [7, 9, 10, 28, 35, 37, 38].

One of the notable extensions to the SCAN method is the translation of the problem to the probabilistic setting. The applications of this problem in probabilistic networks are multiple, for instance, such networks can model transit traffic, with cores representing the most reliably transit stops; such networks can also model protein-protein interactions where cores correspond to proteins that reliably interact with many proteins in the network. The probabilistic version of SCAN, called USCAN [35], introduces the probability of structural similarity for pairs of vertices which finds the probability of their structural similarity being above a threshold ε over all possible worlds. From there, the process continues in similar fashion to SCAN, with vertices added to the structural neighbourhood of a vertex if the probability of their structural similarity is above a threshold η. However, to iterate over all possible worlds would take an exponential amount of time. To combat this, the authors of USCAN devise a Dynamic Programming (DP) method for computing structural similarity. The DP solution runs in quadratic time, however, this is still not practical for computing structural similarity for pairs of vertices with many neighbours. This is indeed the case for most real world networks, where the maximum degree of vertices is well in the millions. The result is that the DP solution is unable to handle pairs of high degree vertices or large networks with many medium to high degree vertices.

Proceedings of the VLDB Endowment, Vol. 16, No. 11 ISSN 2150-8097.
doi:10.14778/3611479.3611516

In this paper, we propose an efficient statistical approximation method for calculating the probability of structural clustering being above a given threshold. Our calculation is built on Lyapunov's version of the Central Limit Theorem (CLT) [43] which works for non-identically distributed but independent random variables. The crux of our method is to express the Jaccard similarity of a pair of vertices in a probabilistic graph as the sum of a special set of random variables which we prove to give the structural similarity of the pair. This needs to be done carefully in order to properly satisfy the highly technical conditions stipulated in the Lyapunov CLT. Once we achieve the expression of structural similarity in terms of a sum of random variables that satisfies Lyapunov CLT, the problem then becomes that of computing a one-tailed area under the Normal Distribution, which is easily done. Our approach runs in linear time with respect to the number of neighbours between a pair of vertices connected by an edge. Since, the CLT is an approximation to the true distribution of the sum of random variables, our approach also yields an approximate solution. However, it is well known that CLT produces a very tight approximation in practice for large numbers. This is also what we observe for our problem. For pairs of vertices with a number of neighbours in the few hundreds, which is where DP starts being impractical, our CLT approach produces approximations that are indistinguishable from numbers produced by DP. We give theoretical bounds on the quality of the approximation using the Berry-Essen Theorem [16, 43].

To reiterate, the complexity of our method reduces to linear time from the quadratic time achieved by DP, with our method running up to three orders of magnitude faster for datasets that DP can handle. Furthermore, our method achieves significantly greater scalability, clustering graphs with up to half a billion edges in less than an hour. Meanwhile USCAN was not able to complete on datasets with more than 30 million edges.

We give in the following a summary of our contributions.

- We derive an efficient approximate method to calculate the probability of structural similarity with the Lyapunov Central Limit Theorem, which gives practically identical results to the exact computation from the Dynamic Programming solution. We give a proof of correctness and bound the quality of the approximation solution.
- We derive the time complexity of our method and validate its time improvement over USCAN through experimentation on real world datasets. We show that our method yields up to three orders of magnitude improvement in time over the exact calculation.
- The reduction in time complexity allows our method to scale up to much larger datasets than the state-of-art USCAN algorithm. Our algorithm finishes in less than an hour on graphs with over half a billion edges.

2 BACKGROUND

Definition 1 (Probabilistic Graph). *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, p)$ be an un-directed probabilistic graph s.t. $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ and $p : \mathcal{E} \longrightarrow (0, 1]$.*

Definition 2 (Possible Worlds). *Unlike deterministic graphs, probabilistic graphs represent possible worlds, which all have different probabilities of occurring. A graph $G = (\mathcal{V}, E)$, where $E \subseteq \mathcal{E}$, is a possible world of $\mathcal{G}$, and the probability of occurring from $\mathcal{G}$ is:*

$$P[G|\mathcal{G}] = \prod_{e \in E} p(e) \prod_{e \in \mathcal{E} \setminus E} (1 - p(e)) \tag{1}$$

and we say $G \sqsubseteq \mathcal{G}$, meaning G is a possible world of $\mathcal{G}$. Hence, for a probabilistic graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E}, p)$, there are $2^{|\mathcal{E}|}$ possible worlds of $\mathcal{G}$, where each edge $e \in \mathcal{E}$ has probability $p(e)$ of existing in a possible world G.

2.1 Definitions

Our method uses the same framework as pSCAN and USCAN [7, 35] which in turn are based on SCAN [42]. Specifically, we present the following definitions.

Definition 3 (Structural Neighbourhood [42]). *Given a deterministic graph $G = (\mathcal{V}, E)$, the structural neighbourhood, N_u, of a vertex $u \in \mathcal{V}$, is a closed neighbourhood, meaning $N_u = \{v \in \mathcal{V} \mid (u, v) \in E\} \cup \{u\}$. That is, the structural neighbourhood of u, contains u by definition.*

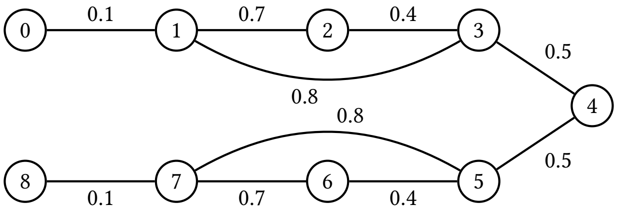

Figure 1: Probabilistic graph example $\mathcal{G} = (\mathcal{V}, \mathcal{E}, p)$. $\overline{G} = (\mathcal{V}, \mathcal{E})$ is the maximal possible world of $\mathcal{G}$, where all $e \in \mathcal{E}$ are present.

Figure 1 is an example of a probabilistic graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E}, p)$, with nine vertices and ten edges. For this graph, $\mathcal{G}$, there are $2^{|\mathcal{E}|} = 2^{10} = 1024$ distinct possible worlds. Let $\overline{G} = (\mathcal{V}, \mathcal{E})$ be the possible world graph where all edges $e \in \mathcal{E}$ are present. We call this possible world the maximal possible world of $\mathcal{G}$. Moreover, we denote the structural neighbourhoods of the maximal possible world as $\overline{N_u}$, $\forall u \in \mathcal{V}$.

Example 1. *Consider vertices 1 and 3 in the deterministic graph $\overline{G} = (\mathcal{V}, \mathcal{E})$ from Figure 1. The structural neighbourhoods of vertices 1 and 3 are $\overline{N_1} = \{0, 1, 2, 3\}$ and $\overline{N_3} = \{1, 2, 3, 4\}$ respectively.*

Definition 4 (Structural Similarity [35]). *Given a deterministic graph $G = (V, E)$, the structural similarity between vertices u and v, $\sigma(u, v)$, is defined as the number of common structural neighbours between u and v, divided by the number of structural neighbours in either u or v, that is*

$$\sigma(u, v) = \frac{|N_u \cap N_v|}{|N_u \cup N_v|} \tag{2}$$

where Equation 2 is the Jaccard similarity, which is an effective measure for structural clustering in networks [7, 35, 37].

Example 2. *Consider the edge $(1, 3)$ in the deterministic graph $\overline{G}$ from Figure 1. The structural neighbourhoods for this edge are given in Example 1. The sizes of the intersection and union are then $|\overline{N_1} \cap \overline{N_3}| = |\{1, 2, 3\}| = 3$ and $|\overline{N_1} \cup \overline{N_3}| = |\{0, 1, 2, 3, 4\}| = 5$, hence $\sigma(1, 3) = \frac{3}{5}$.*

Definition 5 ($\varepsilon-$Structural Similarity [42]). *Given a deterministic graph $G = (V, E)$, an edge $(u, v) \in E$, and threshold ε, u is $\varepsilon-$structural similar to v if $\sigma(u, v) \geq \varepsilon$.*

Example 3. *The $\varepsilon-$structural similarity of the edge $(1, 3)$ is $\sigma(1, 3) = \frac{3}{5}$ in $\overline{G}$ from Figure 1. Hence, if $\varepsilon = \frac{1}{2}$, then vertex 1 is $\varepsilon-$structural similar to 3 and vice versa.*

Thus far, all the definitions have to do with deterministic graphs, and largely come from SCAN [42]. Next we introduce key ideas that hail from USCAN [35] which are designed to elevate the SCAN model to probabilistic networks.

Definition 6 (Probability of Structural Similarity [35]). *Given a similarity threshold $\varepsilon \in (0, 1]$, the probability that $\sigma(e) \geq \varepsilon$ is the sum of the probabilities over all possible worlds $G \sqsubseteq \mathcal{G}$, such that the structural similarity of $e = (u, v)$ is no less than ε in G. That is,*

$$P[e, \varepsilon] = \sum_{G \sqsubseteq \mathcal{G}} P[G|\mathcal{G}] \cdot \Theta(\sigma(e) \geq \varepsilon) \tag{3}$$

where $\Theta(\sigma(e) \geq \varepsilon)$ is an indicator function that equals 1 when $\sigma(e) \geq \varepsilon$, and 0 otherwise.

Example 4. *Consider the edge $(1, 3)$ in the probabilistic graph $\mathcal{G}$ from Figure 1. There are a total of 1024 possible worlds of $\mathcal{G}$, each of which occurs with probability derived from Equation 1 based on the included edges. Suppose that $\varepsilon = \frac{1}{2}$, then only the possible worlds where $\sigma(1, 3) \geq \frac{1}{2}$ contribute to the sum. Using Equation 3, $P\left[(1, 3), \frac{1}{2}\right] = 0.7784$.*

We can now define the notion of reliable neighbourhoods and reliable core vertices using Definition 6.

Definition 7 (Reliable Structural Similarity [35]). *Given an edge e and threshold η, u is reliable structural similar to v if $P[e, \varepsilon] \geq \eta$.*

Example 5. *Consider $\mathcal{G}$ in Figure 1, the probability of structural similarity for edge $(1, 3)$ is $P\left[(1, 3), \frac{1}{2}\right] = 0.7784$. Then if $\eta = \frac{2}{3}$, vertices 1 and 3 are reliable structurally similar to each other since $P\left[(1, 3), \frac{1}{2}\right] \geq \frac{2}{3}$.*

Definition 8 ($(\varepsilon, \eta)-$Reliable Neighbourhood [35]). *Given a similarity threshold $\varepsilon \in (0, 1]$, and a probability threshold $\eta \in (0, 1]$, the $(\varepsilon, \eta)-$reliable neighbourhood of u is the subset of vertices in $\overline{N_u}$ such that $P[(u, v), \varepsilon] \geq \eta$, meaning the set is given by $N_u(\varepsilon, \eta) = \{v \in \overline{N_u} \mid P[(u, v), \varepsilon] \geq \eta\}$.*

Example 6. *When $\eta = \frac{2}{3}$ and $\varepsilon = \frac{1}{2}$, then the $(\varepsilon, \eta)-$reliable neighbourhoods in Figure 1 are: $N_0\left(\frac{1}{2}, \frac{2}{3}\right) = \{0\}$, $N_1\left(\frac{1}{2}, \frac{2}{3}\right) = \{1, 2, 3\}$, $N_2\left(\frac{1}{2}, \frac{2}{3}\right) = \{1, 2\}$, $N_3\left(\frac{1}{2}, \frac{2}{3}\right) = \{1, 3\}$, $N_4\left(\frac{1}{2}, \frac{2}{3}\right) = \{4\}$, $N_5\left(\frac{1}{2}, \frac{2}{3}\right) = \{5, 7\}$, $N_6\left(\frac{1}{2}, \frac{2}{3}\right) = \{6, 7\}$, $N_7\left(\frac{1}{2}, \frac{2}{3}\right) = \{5, 6, 7\}$, $N_8\left(\frac{1}{2}, \frac{2}{3}\right) = \{8\}$.*

Notice that for all $(\varepsilon, \eta)-$reliable neighbourhoods every vertex u is contained in its own $(\varepsilon, \eta)-$reliable neighbourhood $N_u(\varepsilon, \eta)$. Recall each vertex has a minimum structural neighbourhood size of one by the definition, and every vertex is in every possible world. Consider that each node is connected to itself via *self loop*, then $P[(u, u), \varepsilon] = 1$, $\forall \varepsilon$. Therefore, all vertices are in their own $(\varepsilon, \eta)-$reliable neighbourhood, by definition.

Definition 9 ($(\varepsilon, \eta, \mu)-$Reliable Core Vertex [35]). *Given a similarity threshold $\varepsilon \in (0, 1]$, a probability threshold $\eta \in (0, 1]$, and an integer threshold $\mu \geq 2$, a vertex u is a $(\varepsilon, \eta, \mu)-$reliable core vertex if $|N_u(\varepsilon, \eta)| \geq \mu$.*

Example 7. *When $\mu = 3$, and with the $\left(\frac{1}{2}, \frac{2}{3}\right)-$reliable neighbourhoods from the previous example, only vertices 1 and 7 are reliable core vertices; because they are the only nodes with reliable neighbourhoods that contain three or more elements.*

Definition 10 (Reliable Structure-reachable [35]). *Given parameters $\varepsilon \in (0, 1]$, $\eta \in (0, 1]$, and $\mu \geq 2$, vertex v is reliable structure-reachable from vertex u if there is a sequence of vertices $v_1, \ldots, v_l \in V$ with $l \geq 2$, such that:*

- *$v_1 = u$ and $v_l = v$;*
- *$v_1, v_2, \ldots, v_{l-1}$ are reliable core vertices;*
- *$v_{i+1} \in N_{v_i}(\varepsilon, \eta)$ for each $i \in [1, l-1]$*

For v to be reliable structure-reachable from u means there is a path of reliable core vertices from u that reaches v. Notice from the definition, v does not need to be a reliable core vertex. The only requirement of v is that it belongs to the reliable neighbourhood set of the last reliable core vertex in the path of reliable core vertices starting from u.

Example 8. *Consider the probabilistic graph in Figure 1. Let the parameters $(\varepsilon, \eta, \mu) = \left(\frac{1}{2}, \frac{2}{3}, 3\right)$. Since the only core vertices are 1 and 7, which are disconnected from each other, we have that all reliable structure-reachable paths are merely a single edge from the two core vertices to each of their $(\varepsilon, \eta)-$reliable neighbours. Thus, vertices 2 and 3 are reliable structure-reachable from 1; and vertices 5 and 6 are reliable structure-reachable from 7.*

From the definitions above, USCAN [35] formulated the problem of structural clustering on probabilistic graphs as follows.

Definition 11 (The Probabilistic Graph Clustering Problem [35]). *Given a probabilistic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, p)$ and parameters $\varepsilon \in (0, 1]$, $\eta \in (0, 1]$, and $\mu \geq 2$, the problem of probabilistic graph clustering is to compute the set $\mathbb{C}$ of reliable clusters in $\mathcal{G}$. Each reliable cluster, $C \in \mathbb{C}$, must contain at minimum two vertices and satisfy:*

- ***Maximality**: for each reliable core vertex $u \in C$, all vertices that are reliable structure-reachable from u must be in C.*
- ***Connectivity**: for any two vertices $v_1, v_2 \in C$, $\exists\, u \in C$ s.t. both v_1, v_2 are reliable structure-reachable from u.*

Example 9. *Consider the probabilistic graph, $\mathcal{G}$, in Figure 1 with $(\varepsilon, \eta, \mu) = \left(\frac{1}{2}, \frac{2}{3}, 3\right)$. By maximality, and connectivity, the nodes 1,2,3 and 5,6,7 form two distinct clusters.*

Notice with these definitions it is possible for a non-core vertex to belong to multiple clusters at once. Suppose the example graph in Figure 1 had an additional non-core vertex 9, and this vertex was connected to the graph in such a way that 9 is in both the $(\varepsilon, \eta)-$reliable neighbourhoods of 1 and 7. Then vertex 9 would be reliable structure-reachable from both 1 and 7, and therefore would be apart of both the clusters formed in Example 9. Hence the cluster sets produced are not partitions since overlaps are permitted.

Definition 12 (Hubs and Outliers [35]). *Given the set $\mathbb{C}$ of reliable clusters in a probabilistic graph $\mathcal{G}$, a vertex u that is not in any reliable cluster in $\mathbb{C}$ is a hub vertex if it connects two or more reliable clusters, and it is an outlier vertex otherwise.*

It is possible that for a given probabilistic graph, $\mathcal{G}$, no hubs or outliers are found after identifying the set of clusters $\mathbb{C}$ with specified values for the parameters η, ε, and μ.

Example 10. *From Example 9 we have two clusters $C_1 = \{1, 2, 3\}$ and $C_2 = \{5, 6, 7\}$. Vertex 4 is not in any cluster, but is attached by an edge to clusters C_1 and C_2 via vertices 3 and 5 respectively; therefore, vertex 4 is a hub. Additionally, vertices 0, and 8 are also not in any cluster. Unlike 4, vertices 0 and 8 only connect to one cluster each via edges to 1 and 7 respectively. Therefore, vertices 0 and 8 are outliers.*

DP algorithm. To overcome the $O(2^{|\mathcal{E}|})$ complexity for computing the probability of structural similarity, the authors of USCAN derive a clever approach of computing the probability in Equation 3. After all of the optimization observations are applied, the DP algorithm runs in $O(|\overline{N_u} \cup \overline{N_v}|^2)$ time in the worst case [35]. Please see Appendix in the full version for a detailed explanation of the DP method.

2.2 Framework

The framework for producing clusters, hubs, and outliers is inherited from pSCAN [7] and subsequently USCAN [35]. The distinction between our method and the state-of-art resides in the function ComputePr(u, v, ε). In USCAN, ComputePr(u, v, ε) is the DP calculation to determine the probability of structural similarity. For our method, if an edge has a neighbourhood union size that meets a preset threshold parameter, then Lyapunov CLT is used to find the value of $P[(u, v), \varepsilon]$; the time complexity of using Lyapunov CLT is linear in the neighbourhood union size.

Algorithm 1 Clustering Framework

```
1:  procedure Framework($\mathcal{G} = (\mathcal{V}, \mathcal{E}, p)$)
2:      Initialize $G_c = (\mathcal{V}, \emptyset)$
3:      $\forall u \in \mathcal{V}$, initialize $u$ as a non-core vertex
4:      for each $u \in \mathcal{V}$ do
5:          if IsReliableCore($u$) then Label $u$ as a core vertex
6:          for each $v \in N_u(\varepsilon, \eta)$ do
7:              if IsReliableCore($v$) then Add $(u, v)$ to $G_c$
8:      $\mathbb{C}_c \leftarrow$ the set of connected components in $G_c$
9:      $\mathbb{C} \leftarrow \{C_c \cup_{u \in C_c} N_u(\varepsilon, \eta) \mid C_c \in \mathbb{C}_c\}$
10:     return $\mathbb{C}$
11: procedure IsReliableCore($u$)
12:     $N_u(\varepsilon, \eta) \leftarrow \emptyset$
13:     for each $v \in N_u \setminus \{u\}$ do
14:         ComputePr($u, v, \varepsilon$)
15:         if $P[(u, v), \varepsilon] \geq \eta$ then Add $v$ to $N_u(\varepsilon, \eta)$
16:     if $|N_u(\varepsilon, \eta)| \geq \mu$ then return True
17:     else return False
```

Algorithm 1 starts by initializing an edgeless graph G_c with all the vertices in $\mathcal{G}$, line 2. Each vertex becomes marked as a non-core vertex in line 3. Then the algorithm checks whether each vertex is a reliable core vertex, lines 4-5. For each reliable core vertex u found, any reliable neighbours of that vertex that are also reliable core vertices v, have their corresponding edge (u, v) added to G_c, lines 6-7. The graph G_c now exclusively contains edges that connect reliable core vertices together. Thus, the connected components of G_c begin to form the clusters in $\mathbb{C}$, line 8. However, by the definition of reliable structure-reachable, the last vertex in the path need not be a reliable core. Hence, each vertex, u, in each cluster must include their $(\varepsilon, \eta)-$reliable neighbourhood into their cluster as well, line 9.

For Algorithm 1, the proof of correctness is given in [7]. Lines 2-8 take $O(m)$ time, if ComputePr(u, v, ε) is constant. However, the DP method ComputePr(u, v, ε) takes $O(|\overline{N_u} \cup \overline{N_v}|^2)$. From the analysis in Qiu et. al. [35], the entire clustering process takes $O(d_{max}^2 \times \alpha \times m)$, where α is the arboricity of the graph which comes from the original proof in [7], and m is the number of edges in $\mathcal{G}$ [11].

2.3 Challenges

The method proposed in this paper aims to reduce the time complexity of the bottleneck process for clustering probabilistic graphs. In the USCAN algorithm, the process that takes the most time is the DP algorithm that calculates $P[(u, v), \varepsilon]$. The DP algorithm takes $O(|\overline{N_u} \cup \overline{N_v}|^2)$ time for a single edge $(u, v) \in \mathcal{E}$. In our proposed algorithm, our Lyapunov CLT approach computes $P[(u, v), \varepsilon]$ in $O(|\overline{N_u} \cup \overline{N_v}|)$ time.

3 PROPOSED ALGORITHM

We now describe our proposed algorithm, NUSCAN, for computing $P[(u, v), \varepsilon]$. In Section 3.1, we show the core technique of NUSCAN, which makes use of Lyapunov Central Limit Theorem for computing $P[(u, v), \varepsilon]$. Then, we decribe the main steps of NUSCAN in Section 3.2. Finally, in Section 3.3, we derive bounds on the quality of the solution for NUSCAN.

3.1 Structural Similarity using Lyapunov CLT

Theorem 1. *[Lyapunov CLT] Let $\xi_1, \xi_2, \dots, \xi_n$ be a sequence of independent, but non-identically distributed random variables, each with finite expected value μ_k and variance σ_k^2. Let*

$$s_n^2 = \sum_{k=1}^{n} \sigma_k^2 \tag{4}$$

Lyapunov CLT states if

$$\lim_{n \to \infty} \frac{1}{s_n^{2+\delta}} \sum_{k=1}^{n} E[|\xi_k - \mu_k|^{2+\delta}] = 0 \tag{5}$$

for some $\delta > 0$, then $\frac{1}{s_n} \sum_{k=1}^{n} (\xi_k - \mu_k)$ converges in distribution to a standard normal random variable [12, 16, 43].

The limit condition in Equation 5 is difficult to show analytically. In Cuzzocrea et. al. [12], they prove a sufficient condition for Equation 5 in Lyapunov CLT to holds.

Theorem 2. *[12] Given a sequence of independent random variables $\{\xi_k, k = 1, \dots, n\}$ such that $E[(\xi_k - \mu_k)^2] = \sigma_k^2 > 0\ \forall k$ holds and the centered 3-rd moments $E[|\xi_k - \mu_k|^3] = \eta_k < \infty\ \forall k$, then the Lyapunov CLT limit condition in Equation 5 holds.*

Remark. Showing that Lyapunov CLT can be applied to calculate the probability of structural similarity is technical and requires care. Suppose that $J_{u,v}$ is the random variable that represents the value of

$\sigma(u,v)$ over all possible worlds. Then the calculation of $P[(u,v),\varepsilon]$ becomes equivalent to the expression $P[J_{u,v} \geq \varepsilon] \times p(u,v)$. We show that the random variable $J_{u,v}$ can be expressed as the sum of independent, non-identically distributed random variables. In what follows, we put forth a series of definitions and lemmas needed to derive an expression for the probability of structural similarity, $P[(u,v),\varepsilon]$, allowing the use of Lyapunov CLT.

To start, we define for each edge in the probabilistic graph a Bernoulli Random Variable that indicates whether a given edge is present in any possible world according to its edge probability.

Definition 13 (Edge Random Variable). *Given a probabilistic graph $\mathcal{G} = (\mathcal{V},\mathcal{E},p)$, let $X_{u,v}$ be a Bernoulli Random Variable that determines whether an edge $(u,v) \in \mathcal{E}$ is present in an arbitrary possible world G, meaning*

$$P[X_{u,v} = 1] = p(u,v) \text{ and } P[X_{u,v} = 0] = 1 - p(u,v) \tag{6}$$

where $p(u,v)$ is the probability that edge (u,v) is in any possible world $G \sqsubseteq \mathcal{G}$. We call $X_{u,v}$ an Edge Random Variable (ERV).

The sequence of all ERV is by definition a sequence of independent and non-identically distributed random variables since each ERV may have a different value of $p(u,v)$.

Example 11. *Consider the probabilistic graph in Figure 1, and specifically the edge $(1,3)$. The probability that $(1,3)$ is in any arbitrary possible world G is $p(1,3) = 0.8$. Then $X_{1,3}$ has the value 1 with probability 0.8 and is 0 otherwise.*

Next, we construct a special sequence of vertices from the combined neighbourhood sets between an edge (u,v). We will use this sequence in order to derive the distribution of $J_{u,v}$.

Definition 14 (Neighbourhood Edge Sequence). *Given a probabilistic graph $\mathcal{G} = (\mathcal{V},\mathcal{E},p)$, $\forall (u,v) \in \mathcal{E}$ let $N^*_{uv} = (\overline{N_u} \cap \overline{N_v}) \setminus \{u,v\}$ be the set of common neighbours excluding the vertices $\{u,v\}$ and let $\widetilde{N_{uv}} = (\overline{N_u} \cup \overline{N_v}) \setminus \{u,v\}$ be the set of all neighbours between u and v while excluding $\{u,v\}$. Let Y_{uv} be an ordered sequence of the elements in $\widetilde{N_{uv}}$ such that $y_{2i} = y_{2i+1}$ $\forall i \in [0,q-1]$, where $q = q_{u,v} = |N^*_{uv}|$ and $\forall i \in [0,2q-1]$, $y_i \in N^*_{uv}$; and $\forall j \in [2q,r-1]$, $y_j \in \widetilde{N_{uv}} \setminus N^*_{uv}$ where $r = |\widetilde{N_{uv}}| + |N^*_{uv}|$. Therefore,*

$$Y_{uv} : y_0, y_1, \ldots, y_{2q-1}, y_{2q}, \ldots, y_{r-1} \tag{7}$$

Without loss of generality, the elements in $\overline{N_u} \setminus \overline{N_v}$ appear in the sequence before the elements that are in $\overline{N_v} \setminus \overline{N_u}$.

For each edge (u,v) in a probabilistic graph, there is an associated Neighbourhood Edge Sequence Y_{uv} with three distinct sections. The first section contains the elements that are in the maximal neighbourhoods of both u,v (excluding u, and v themselves). The elements in the first section are duplicated to signify membership to both maximal neighbourhoods. The second section of the sequence contains elements exclusively belonging to the maximal neighbourhood of u. The third section of Y_{uv} holds elements only in the maximal neighbourhood of v. In the second and third sections, the elements only appear once as opposed to the first section where elements are repeated. The reason is to symbolized ownership of the vertex to only one maximal neighbourhood set; contrast to the first section where the represented vertex belonged to both maximal neighbourhood sets. It is possible that any of the three sections of Y_{uv} do not contribute any elements. The three sections of Y_{uv} discussed above derive from three sets N^*_{uv}, $\overline{N_u} \setminus \overline{N_v}$, and $\overline{N_v} \setminus \overline{N_u}$ respective to the outlined order above, which for some $(u,v) \in \mathcal{E}$ may be empty. Therefore, for some (u,v) if $N^*_{uv} = \overline{N_u} \setminus \overline{N_v} = \overline{N_v} \setminus \overline{N_u} = \emptyset$, then Y_{uv} is an empty sequence.

Example 12. *Consider the probabilistic graph in Figure 1, and the edge $(1,3)$. In $\overline{G}$, the edge $(1,3)$ has maximal structural neighbourhoods $\overline{N_1} \setminus \{1,3\} = \{0,2\}$ and $\overline{N_3} \setminus \{1,3\} = \{2,4\}$. Then $\widetilde{N_{13}} = \{0,2,4\}$ and $N^*_{13} = \{2\}$, and thus Y_{13} is 2, 2, 0, 4.*

From Y_{uv}, we define a homomorphic sequence of Edge Random Variables for the edges represented in the original sequence.

Definition 15 (Correspondence Sequence). *Given $\mathcal{G} = (\mathcal{V},\mathcal{E},p)$, $\forall (u,v) \in \mathcal{E}$ with Neighbourhood Edge Sequence Y_{uv}, let χ_{uv} be a sequence of ERV in one-to-one correspondence to Y_{uv} under the following definition,*

$$\chi_{uv} : X_{y_0,u}, X_{y_1,v}, \ldots, X_{y_{2q-2},u}, X_{y_{2q-1},v}, X_{y_{2q},z}, \ldots, X_{y_{r-1},z} \tag{8}$$

where $X_{y_i,z}$ is the ERV for the edge (y_i,z), and z is either u or v as defined by Y_{uv}.

Example 13. *Suppose we have the sequence Y_{13} : 2, 2, 0, 4. Then the Correspondence Sequence is $\chi_{13} : X_{2,1}, X_{2,3}, X_{0,1}, X_{4,3}$.*

Unlike Y_{uv} where elements were integers and not necessarily unique, each element of χ_{uv} is a unique random variable that represents the same corresponding edge in Y_{uv}. Now for each edge (u,v), we have a sequence of ERV that represents edges in both maximal neighbourhoods of u and v. We exploit the ordering of this sequence to derive two random variables that constitute the numerator and denominator of $J_{u,v}$.

Lemma 1. *Given a probabilistic graph $\mathcal{G} = (\mathcal{V},\mathcal{E},p)$, $(u,v) \in \mathcal{E}$, and sequence χ_{uv}, the random variable that represents $|N_u \cap N_v|$ over all possible worlds is defined as,*

$$\mathcal{M}_{u,v} = 2 + \sum_{i=0}^{q-1} X_{y_{2i},u} X_{y_{2i+1},v} \tag{9}$$

Proof. The 2 is for the presence of u,v, and the sum contributes elements possibly in the intersection when both ERV equal 1. □

Example 14. *In Example 13, the Correspondence Sequence for edge $(1,3)$ was $\chi_{13} : X_{2,1}, X_{2,3}, X_{0,1}, X_{4,3}$; then $\mathcal{M}_{1,3} = 2 + X_{2,1}X_{2,3}$.*

Lemma 2. *Given a probabilistic graph $\mathcal{G} = (\mathcal{V},\mathcal{E},p)$, $(u,v) \in \mathcal{E}$, and sequence χ_{uv}, the random variable that represents $|N_u \cup N_v|$ over all possible worlds is defined as,*

$$\mathcal{N}_{u,v} = 2 + \sum_{i=0}^{q-1} \max\left(X_{y_{2i},u}, X_{y_{2i+1},v}\right) + \sum_{\substack{j=2q \\ z \in \{u,v\}}}^{r-1} X_{y_j,z} \tag{10}$$

Proof. The 2 is once again for the presence of u, and v. The first sum counts intersecting elements if at least one of the ERV is equal to 1. The second sum counts elements outside the intersection when their ERV are equal to 1. □

Example 15. *In Example 13, the Correspondence Sequence was $\chi_{13} : X_{2,1}, X_{2,3}, X_{0,1}, X_{4,3}$; so $\mathcal{N}_{1,3} = 2 + \max(X_{2,1}, X_{2,3}) + X_{0,1} + X_{4,3}$.*

For edge (u,v), the random variables $\mathcal{M}_{u,v}$ and $\mathcal{N}_{u,v}$ represent respectively, the size of the intersection and union of structural neighbourhoods over all possible worlds. The two random variables $\mathcal{M}_{u,v}$ and $\mathcal{N}_{u,v}$ assume that the edge u, and v exist. Hence any further derived random variables inherit this assumption of existence. Using the random variables $\mathcal{M}_{u,v}$ and $\mathcal{N}_{u,v}$, we derive the probabilistic Jaccard similarity $J_{u,v}$.

Corollary 1. *Given a probabilistic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, p)$, $(u, v) \in \mathcal{E}$, sequence χ_{uv}, and random variables $\mathcal{M}_{u,v}$ and $\mathcal{N}_{u,v}$, the Jaccard similarity over all possible worlds is,*

$$J_{u,v} = \frac{\mathcal{M}_{u,v}}{\mathcal{N}_{u,v}} \tag{11}$$

Proof. Lemmas 1 and 2 proved that $\mathcal{M}_{u,v}$ and $\mathcal{N}_{u,v}$ are the random variable representation of the intersection and union of the structural neighbourhoods of u, and v over all possible worlds. Therefore it follows the ratio of $\mathcal{M}_{u,v}$ to $\mathcal{N}_{u,v}$ is exactly the random variable representation of $\sigma(u, v)$ over all possible worlds. □

Example 16. *From Examples 14 and 15, $\mathcal{M}_{1,3} = 2 + X_{2,1}X_{2,3}$ and $\mathcal{N}_{1,3} = 2 + \max(X_{2,1}, X_{2,3}) + X_{0,1} + X_{4,3}$. Therefore, for edge $(1, 3)$, the probabilistic Jaccard similarity is,*

$$J_{1,3} = \frac{\mathcal{M}_{1,3}}{\mathcal{N}_{1,3}} = \frac{2 + X_{2,1}X_{2,3}}{2 + \max(X_{2,1}, X_{2,3}) + X_{0,1} + X_{4,3}}$$

We now have a random variable representation of the structural similarity measure $\sigma(u, v)$ over all possible worlds, called the probabilistic Jaccard similarity $J_{u,v}$. Next we determine the probability that $J_{u,v} \geq \varepsilon$, where $\varepsilon \in (0, 1]$.

$$P[J_{u,v} \geq \varepsilon] = P\left[\frac{\mathcal{M}_{u,v}}{\mathcal{N}_{u,v}} \geq \varepsilon\right] = P[\mathcal{M}_{u,v} - \varepsilon\mathcal{N}_{u,v} \geq 0] \tag{12}$$

In order to approximate $P[J_{u,v} \geq \varepsilon]$, we wish to employ the Lyapunov CLT. Before we proceed, the random variables must be independent. Since $\mathcal{M}_{u,v}$, $\mathcal{N}_{u,v}$ contain some overlapping random variables in their definitions, they are not independent. So we substitute in the formulas for $\mathcal{M}_{u,v}$ and $\mathcal{N}_{u,v}$ to decouple the ERV in the sum over the first $2q - 1$ terms. Then Equation 12 becomes,

$$\begin{aligned} &= P\Big[2(1-\varepsilon) + \sum_{i=0}^{q-1} \Big\{X_{y_{2i},u}X_{y_{2i+1},v} - \varepsilon \max(X_{y_{2i},u}, X_{y_{2i+1},v})\Big\} \\ &\quad - \varepsilon \sum_{\substack{j=2q \\ z\in\{u,v\}}}^{r-1} X_{y_j,z} \geq 0\Big] \end{aligned} \tag{13}$$

The term inside the first summand depends on two ERV $X_{y_{2i},u}$, and $X_{y_{2i+1},v}$, which combined have four possible outcomes. We derive a new random variable that encapsulates all possible states of the expression inside the first sum of Equation 13.

Proposition 1. *Given a probabilistic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, p)$, $(u, v) \in \mathcal{E}$, sequence χ_{uv}, for $i \in [0, q-1]$, let $Z(u, v, y_{2i})$ be a random variable such that $Z(u, v, y_{2i}) = X_{y_{2i},u}X_{y_{2i+1},v} - \varepsilon \max(X_{y_{2i},u}, X_{y_{2i+1},v})$ then the possible states of $Z(u, v, y_{2i})$ are*

$$\begin{aligned} P[Z(u, v, y_{2i}) = -\varepsilon] &= p_2(1 - p_1) + p_1(1 - p_2) &&= \alpha \quad (14) \\ P[Z(u, v, y_{2i}) = 0] &= (1 - p_1)(1 - p_2) &&= \beta \quad (15) \\ P[Z(u, v, y_{2i}) = 1 - \varepsilon] &= p_1p_2 &&= \gamma \quad (16) \end{aligned}$$

where $p_1 = p(y_{2i}, u)$ and $p_2 = p(y_{2i}, v)$. We call $Z(u, v, y_{2i})$ the Intersect Random Variable.

Notice, $Z(u, v, y_{2i})$ are independent random variables since each one is dependent on distinct pairs of edge random variables. We also rewrite the second sum of Equation 13 into a new random.

Proposition 2. *Given $\mathcal{G} = (\mathcal{V}, \mathcal{E}, p)$, $(u, v) \in \mathcal{E}$, and χ_{uv}, for $i \in [2q, r-1]$, let $W(z, y_i)$ be a random variable such that $W(z, y_i) = (-\varepsilon)X_{y_i,z}$, then the possible states of $W(z, y_i)$ are*

$$P[W(z, y_i) = -\varepsilon] = p_0 \tag{17}$$

$$P[W(z, y_i) = 0] = 1 - p_0 \tag{18}$$

where $p_0 = p(y_i, z)$.

For simplicity of notation, we let $p_1 = p(y_{2i}, u), p_2 = p(y_{2i}, v)$, and $p_0 = p(y_1, z)$. Hence, each $Z(u, v, y_{2i})$ and $W(z, y_i)$ has its own values for p_1, p_2 and p_0 based on the different ERV it represents.

The sets of random variables $Z(u, v, y_{2i})$ and $W(z, y_i)$ are independent but non-identically distributed, as required for the Lyapunov CLT. Let Z be the sum of $Z(u, v, y_{2i})$ Intersect Random Variables; and let W be the sum of $W(z, y_i)$ random variables.

$$Z = \sum_{i=0}^{q-1} Z(u, v, y_{2i}) \text{ and } W = \sum_{\substack{i=2q \\ z\in\{u,v\}}}^{r-1} W(z, y_i) \tag{19}$$

With Z and W, the probability expression $P[J_{u,v} \geq \varepsilon]$ becomes,

$$P[Z + W \geq 2(\varepsilon - 1)] = P[V \geq 2(\varepsilon - 1)] \tag{20}$$

where $V = Z + W$. Therefore we now have a probability expression of independent but non-identically distributed random variables, which satisfies the first condition required for Lyapunov CLT.

Theorem 3. *For the Lyapunov CLT, let $Z(u, v, y_0), Z(u, v, y_2), \ldots, Z(u, v, y_{2q-2}), W(z, y_{2q}), \ldots, W(z, y_{r-1})$ be a sequence of independent but non-identically distributed random variables, each with finite expected value $\mu_{Z(u,v,y_{2i})}$, $\mu_{W(z,y_i)}$ and variance $\sigma^2_{Z(u,v,y_{2i})}$, $\sigma^2_{W(z,y_i)}$. Let the mean and variance be,*

$$\mu_V = \sum_{i=0}^{q-1} \mu_{Z(u,v,y_{2i})} + \sum_{i=2q}^{r-1} \mu_{W(z,y_i)} = \sum_{k=1}^{n} \mu_{V_k} \tag{21}$$

$$s_n^2 = \sum_{i=0}^{q-1} \sigma^2_{Z(u,v,y_{2i})} + \sum_{i=2q}^{r-1} \sigma^2_{W(z,y_i)} = \sum_{k=1}^{n} \sigma^2_{V_k} \tag{22}$$

where $n = r - q$ and V_k is either $Z(u, v, y_{2i})$ or $W(z, y_i)$. Then V converges to a standard normal random variable.

Proof. By definition, V is the sum of independent random variables Z and W. To prove the limit in Equation 5 converges, we use the sufficient condition from Theorem 2. The mean and variance of each $Z(u, v, y_{2i})$ and $W(z, y_i)$ are derived from the definitions,

$$\begin{aligned} E[Z(u, v, y_{2i})] &= \gamma(1 - \varepsilon) - \varepsilon\alpha &&= \mu_Z \quad (23) \\ E[W(z, y_i)] &= -p_0\varepsilon &&= \mu_W \quad (24) \\ E[(Z(u, v, y_{2i}) - \mu_Z)^2] &= \varepsilon^2\alpha + (1 - \varepsilon)^2\gamma - \mu_Z^2 &&= \sigma_Z^2 \quad (25) \\ E[(W(z, y_i) - \mu_W)^2] &= p_0\varepsilon^2(1 - p_0) &&= \sigma_W^2 \quad (26) \end{aligned}$$

where α, β, γ, and p_0 are defined in Equations 14, 15, 16 and Proposition 2 respectively. Let $p_0, p_1, p_2, \varepsilon \in (0, 1)$. Using Equation 23, Equation 25 simplifies to

$$\begin{aligned} \sigma_Z^2 &= \varepsilon^2\alpha + (1 - \varepsilon)^2\gamma - (\gamma(1 - \varepsilon) - \varepsilon\alpha)^2 \\ &= (1 - \varepsilon)^2\gamma(1 - \gamma) + \varepsilon^2\alpha(1 - \alpha) + 2\gamma\alpha\varepsilon(1 - \varepsilon) > 0 \end{aligned} \tag{27}$$

since $\alpha = \alpha(p_1, p_2), \gamma(p_1, p_2)$ are probabilities, and hence $\alpha, \gamma \in (0, 1)$. Note that Equation 26 is strictly greater than zero for the given range of p_0 and ε. Thus the first criterion in Theorem 2 is satisfied. The centred 3rd moments of $Z(u,v,y_{2i})$ and $W(z,y_i)$ are calculated and simplify to,

$$E[|Z(u,v,y_{2i}) - \mu_Z|^3] = |\mu_Z|^3\beta + |\varepsilon + \mu_Z|^3\alpha + |1-\varepsilon-\mu_Z|^3\gamma < \infty \tag{28}$$

$$E[|W(z,y_i) - \mu_W|^3] = |\varepsilon + \mu_W|^3 p_0 + |\mu_W|^3(1-p_0) < \infty \tag{29}$$

Both Equations 28 and 29 are functions of variables with positive degree in the range $(0,1)$. Thus, no points in the allowed domains cause either function to approach infinity. Therefore, V satisfies the requirements of Theorem 2, so the Lyapunov condition holds. □

The probability expression in Equation 20 can now be manipulated such that the Normal Distribution applies.

$$P[V \geq 2(\varepsilon-1)] = Pr[V - \mu_V \geq 2(\varepsilon-1) - \mu_V]$$

$$= P\left[\frac{1}{s_n}\sum_{k=1}^{n} V_i - \mu_{V_k} \geq \frac{1}{s_n}\left(2(\varepsilon-1) - \sum_{k=1}^{n}\mu_{V_k}\right)\right] \tag{30}$$

Therefore, the probability of structural similarity is approximated by the one-tailed area under the Normal Distribution. That is,

$$P[(u,v),\varepsilon] \approx P[V \geq 2(\varepsilon-1)] \times p(u,v) \tag{31}$$

In the following Section, we take the theory developed from this Section and design an algorithm which implements the calculation of the probability of structural similarity as defined in Equation 31.

3.2 NUSCAN

We call our algorithm NUSCAN where "N" is to emphasize the use of the Normal Distribution as per Lyapunov CLT. More specifically, if $|\widetilde{N_{uv}}| \geq t$, then we use Normal Distribution to compute $P[(u,v),\varepsilon]$, for some large $t \in \mathbb{N}$. In practice setting $t = 100$ works well for all graphs (see the full version of the paper located on our GitHub repository for more details). Based on Equations 30 and 31, we propose the following algorithm to compute $P[(u,v),\varepsilon]$ with Lyapunov CLT.

Algorithm 2 Calculation of $P[(u,v),\varepsilon]$

```
1:  procedure ComputePr($u, v, \varepsilon$)
2:      if $p(u,v) < \eta$ then return 0
3:      else if $|\widetilde{N_{uv}}| < t$ then
4:          Use USCAN DP protocol
5:      else
6:          Arrange all $w \in \widetilde{N_{uv}}$ as sequence $Y_{uv}$ (Equation 7)
7:          Split $Y_{uv}$ into $W$ and $Z$ (Equation 19)
8:          $\mu_V \leftarrow 0$, $s_n^2 \leftarrow 0$ (Equations 21 and 22)
9:          for $y_i$ in $W$ do
10:             $\mu_V \leftarrow \mu_V + \mu_{W(z,y_i)}$
11:             $s_n^2 \leftarrow s_n^2 + \sigma^2_{W(z,y_i)}$
12:         for $y_{2i}$ in $Z$ do
13:             $\mu_V \leftarrow \mu_V + \mu_{Z(u,v,y_{2i})}$
14:             $s_n^2 \leftarrow s_n^2 + \sigma^2_{Z(u,v,y_{2i})}$
15:         Let $F_n \leftarrow Norm(\mu_V, s_n)$
16:         return $P\left[F_n \geq \frac{2(\varepsilon-1)-\mu_V}{s_n}\right] \times p(u,v)$ (Equation 31)
```

In Algorithm 2, line 2 is a pruning condition inherited from USCAN. Line 3 checks whether there are enough neighbours in $\widetilde{N_{uv}}$ for the application of the NUSCAN approximation, which is done in constant time. The next part on lines 6-7 prepares the neighbours into two sets Z and W which contain the random variables $Z(u,v,y_{2i})$ and $W(z,y_i)$ respectively, taking only $O(|\overline{N_u} \cup \overline{N_v}|)$. Lines 9-14 calculate and sum the means and variances of each random variable in the sequence as described in Section 3.1, which finishes both loops in $O(|\overline{N_u} \cup \overline{N_v}|)$. In line 15, a Normal Distribution is constructed with mean μ_V and standard deviation s_n, done in constant time. Finally on line 16, the Normal Distribution is used to return the probability that approximates $Pr[(u,v),\varepsilon]$, also in constant time. Then for an edge (u,v), all neighbours in the union are iterated over. Therefore in the worst case, the run time of Algorithm 2 is $O(|\overline{N_u} \cup \overline{N_v}|)$.

Run Time Complexity. We know that the computation of $P[e,\varepsilon]$ is the bottleneck process of the clustering framework. The worst case run times of the DP method and the Lyapunov CLT are $O(|\overline{N_u} \cup \overline{N_v}|^2)$ and $O(|\overline{N_u} \cup \overline{N_v}|)$ respectively on an arbitrary edge $(u,v) \in \mathcal{E}$ in a probabilistic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, p)$. Let $d_{max} = \max_{u \in \mathcal{V}} d(u)$, where $d(u)$ is the number of edges connected to u. Since $|\overline{N_u} \cup \overline{N_v}|$ is bounded above by $2 \times d_{max}$, then $\forall\, e \in \mathcal{E}$, $\exists\, e$ $s.t.$ the DP method takes $O(d_{max}^2)$ and the Lyapunov CLT takes $O(d_{max})$ to compute $P[e,\varepsilon]$. In NUSCAN the edges are partitioned such that we have two edge sets: $E_{DP} = \{e = (u,v) \mid e \in E\ s.t. = |\overline{N_u} \cup \overline{N_v}| < t\}$ and $E_{LCLT} = \{e = (u,v) \mid e \in E\ s.t. = |\overline{N_u} \cup \overline{N_v}| \geq t\}$ where all the edges in E_{DP} have $P[e,\varepsilon]$ computed using the DP method, and the edges in E_{LCLT} use the Lyapunov CLT to compute $P[e,\varepsilon]$. Let $m_D = |E_{DP}|$ and let $m_L = |E_{LCLT}|$, and the maximum $|\overline{N_u} \cup \overline{N_v}|$ in E_{DP} and E_{LCLT} are bounded above by t and $2 \times d_{max}$ respectively. Thus the total runtime of the NUSCAN clustering framework is $O\left(\alpha\left[m_D t^2 + m_L d_{max}\right]\right)$, where α is the arboricity of the graph.

Memory Complexity. Algorithm 2 needs to maintain the neighbourhood intersection and union sets for the current edge, which consumes $O(|\overline{N_u} \cup \overline{N_v}|) = O(d_{max})$ space for the edges in E_{LCLT} and $O(t^2)$ for the edges in E_{DP} in the worst case. The framework algorithm may release this memory for each edge calculation, making the entire memory usage $O(\max(d_{max}, t^2) + m)$ in the worst case. For most practical graphs the memory required is much smaller than the worst case, since for most edges (u,v), $|\overline{N_u} \cup \overline{N_v}|$ is much smaller than d_{max}.

3.3 Approximation Bound

In this Section we bound the error of the Normal Distribution in Equation 30 by using the Berry-Essen Theorem [16, 43].

Theorem 4. *[Berry-Essen Theorem] Given a sequence* $\Gamma_1, \ldots, \Gamma_n$ *of non-identically distributed and independent random variables with* $E[\Gamma_i] = 0$ *and* $E[\Gamma_i^2] = \lambda_i^2$ *and* $E[|\Gamma_i^3|] = \rho_i < \infty$, $\exists$ $C_0 = 0.56$ *s.t. the following is satisfied:*

$$\sup_{x \in \mathbb{R}} |F_n(x) - \Phi(x)| \leq C_0 \left(\sum_{i=1}^{n} \lambda_i^2\right)^{-\frac{3}{2}} \sum_{i=1}^{n} \rho_i \tag{32}$$

where $F_n(x)$ *is the Cumulative Distribution Function (CDF) for*

$$S_n = \frac{\Gamma_1 + \cdots + \Gamma_n}{\sqrt{\lambda_1^2 + \cdots + \lambda_n^2}} \tag{33}$$

C_0 *was determined to be* 0.56 *from previous works [12, 16].*

Theorem 5. *[12] If a set of random variables $\{X_k\}_{k=1}^{n}$ with means μ_k and variances σ_k^2 and centred 3rd moments $E[|X_k - \mu_k|^3]$ s.t. $\sigma_k^2 > 0$, $\forall k$ and $E[|X_k - \mu_k|^3] < \infty$, $\forall k$ then the CDF of Equation 33 converges uniformly to a standard normal CDF, when $\Gamma_k = X_k - \mu_k$.*

We give the following corollary that depicts how to obtain an upper bound on the maximal error of the approximation of V to a Normal Distribution.

Corollary 2. *For edge $(u, v) \in \mathcal{E}$ in $\mathcal{G}$ with random variables $V_1, \ldots, V_n$, the error on the approximation of the right hand side of Equation 30 to the Normal Distribution is given by:*

$$\sup_{x \in \mathbb{R}} |F_n(x) - \Phi(x)| \leq \frac{0.56}{\sqrt{\sum_{i=0}^{q-1} \sigma^2_{Z(u,v,y_{2i})} + \sum_{i=2q}^{r-1} \sigma^2_{W(z,y_i)}}} \quad (34)$$

where $F_n(x)$ is the CDF from Equation 33 when $\Gamma_k = V_k - \mu_{V_k}$.

Proof. We showed in the proof of Theorem 3 that $\sigma_k^2 > 0$ and that the 3rd moments were less than infinity. The proof then follows directly from the application of Theorem 4 and 5. □

4 EXPERIMENTS

In this Section we demonstrate the efficiency, scalability, accuracy, and effectiveness of our proposed algorithm NUSCAN, compared to the state-of-art algorithm USCAN [35]. All algorithms are implemented in C++ and compiled with g++ using the -O3 optimization flag. The experiments are executed on a commodity machine with Intel Xeon E5620, 2.395GHz CPU, and 64Gb RAM, running Ubuntu 18.04. The implementation is available at https://github.com/JoetheManHowie/nuscan.

4.1 Datasets and Experimental Framework

Table 1: Datasets with real or obfuscation probabilities. ρ is the density; C is the cluster coefficient; k and m are the metric scales kilo(10^3), and milli(10^{-3}).

datasets	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	d_{max}	d_{ave}	ρ	C
core	3k	7k	141	5	1.9m	.390
CARoad	1,964k	3,036k	213	3	.002m	.078
douban	87k	157k	222	4	.042m	.015
Flickr	22k	135k	401	12	.557m	.593
DBLP	660k	1,738k	554	5	.008m	.608
biomine	1,008k	6,743k	139,624	13	.013m	.016

Table 2: Datasets are retrieved from Laboratory of Web Algorithmics (https://law.di.unimi.it/datasets.php. M is Mega(10^6).

datasets	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	d_{max}	d_{ave}	ρ	C
enron	.07M	.25M	2k	7	.106m	.14
cnr-2000	.33M	3M	18k	17	.052m	.016
uk-2014-tpd	1.8M	15M	64k	17	.010m	.076
eu-2005	.86M	16M	69k	37	.043m	.029
dewiki-2013	1.5M	33M	118k	44	.029m	.010
eswiki-2013	.97M	21M	145k	44	.045m	.005
uk-2002	18M	262M	195k	28	.002m	.067
indochina-2004	7.4M	151M	256k	41	.001m	.318
arabic-2005	23M	554M	576k	49	.002m	.102

For experimentation we used a combination of real world probabilistic graphs, and deterministic graphs that have probabilities induced from a distribution. As real world graphs, we have *douban, CARoad, core, Flickr, DBLP, and biomine* [15, 26, 29]. *CARoad* and *douban* are datasets from Ma et. al. [29] with the probabilities generated using obfuscation [4] that we received directly from the authors. The *core* dataset comes from Krogan et. al. [26], which represents the uncertainty in protein interaction measurements. The remaining three datasets are from Bonchi et. al. [5]. For the deterministic graphs, all nine were retrieved from Laboratory of Web Algorithmics, and distributions were induced on the edges to create new probabilistic graphs.

Preprocessing. The datasets we use in our experimentation have their statistics given in Tables 1 and 2, and are ordered by the maximum degree. For the algorithms to run properly, all self-loops and isolated nodes are removed from the original datasets. Additionally, directed graphs are converted to undirected graphs by adding symmetrical edges whenever they are missing. The statistics in Tables 1 and 2 reflect the datasets after these modifications. The 15 datasets listed above all underwent the same preprocessing procedure. We note that all the real world datasets did not contain probabilities equal to 0 or 1 nor did they contain self loops. For the deterministic graphs, we induced the probabilities using the power law distribution with $\beta = 2$. For three of the deterministic graphs, *enron, cnr-2000, and uk-2014-tpd*, we additionally generated probabilities using the normal distribution with $\mu = 0.5$ and $\sigma = 0.1$, as well as using the uniform distribution. The programs for executing this procedure are outlined on our GitHub repository which references the specific scripts used. Each dataset runs on 55 different parameter points in the phase space (η, ε, μ) in order to analyse how the variation in parameter values effect the efficiency, scalability, accuracy, and effectiveness of the algorithms.

Goals. Through the following series of experiments we aim to demonstrate that NUSCAN: *a)* gives a highly accurate approximation to USCAN, yielding virtually indistinguishable clusters; *b)* improves the speed of the state-of-art algorithm USCAN by several orders of magnitude; *c)* scales to datasets with over half a billion edges, while USCAN is unable to finish on datasets with more than 30 million edges. In sum, our goal is to show NUSCAN produces cluster sets near identical to USCAN in a fraction of the time. For experimentation, we use the real world datasets to compare between USCAN in terms of cluster set results; whereas the nine larger datasets demonstrate the scalability of NUSCAN.

4.2 Comparison to USCAN

To analyze the differences in clustering results between USCAN and NUSCAN, we observe both the global clustering output and the local calculations of $P[e, \varepsilon]$.

Cluster Comparison. After running USCAN and NUSCAN on the real world graphs, the sets of clusters, hubs, outliers, and core vertices were found and saved. To identify the agreement between the two cluster sets, we measured the average Jaccard similarity between pairs of clusters that shared more than half of their elements. The clusters that did not have matches were less than about 1%, and were very small in size.

For the sets of hubs, outliers, and core vertices we also compute the Jaccard similarity. On all six of the real world graphs, we found that on the overwhelming majority of parameter points, the results from NUSCAN were identical to USCAN. Specifically, for *douban* *and CARoad* it was only the results for the three points (0.2, 0.1, 2), (0.5, 0.1, 2), and (0.5, 0.1, 5) that did not match completely. Yet even for those points, the results still showed more than 99% match for cluster and core sets, and more than 90% for the other sets (hubs and outliers).

In the other four real world graphs, *core, Flickr, DBLP, and biomine*, the majority of points produced perfect cluster matches to USCAN; with the remaining points matching the clusters, cores, hubs, and outliers with at least 90% accuracy. Table 3 displays the matching fractions for the point (0.5, 0.2, 2) on all six graphs. So to summarize, we found that for all the 55 parameter points the results from NUSCAN and USCAN were in near perfect agreement, with the cluster set having more than 99% match in every case, and the sets of hubs, cores, and outliers matching in range between 90% to 100% with a strong tendency to be at the upper end of that range.

Calculation of $P[e, \varepsilon]$. For a local comparison of the two clustering algorithms, we analyze the key difference between them, which is the computation of $P[e, \varepsilon]$. In USCAN, all $P[e, \varepsilon]$ values are computed with DP; whereas in NUSCAN, $P[e, \varepsilon]$ values are computed with Lyapunov CLT when the number of total distinct neighbours for e is larger than 100. To measure the error between the two methods we calculated the root mean squared error (*RMSE*).

Table 3: Jaccard similarity of the cluster sets, core sets, hub sets, and outlier sets between USCAN and NUSCAN. S is the set of edges that pass through Lyapunov method in NUSCAN. RMSE is measured for edges in $|S|$. $(\eta, \varepsilon, \mu) = (0.5, 0.2, 2)$.

datasets	$\|S\|$	RMSE	clusters	cores	hubs	outliers
douban	2	.080	1.0	1.0	1.0	1.0
CARoad	6	.002	1.0	1.0	1.0	1.0
core	8	.161	1.0	.986	.997	.999
Flickr	37	.143	1.0	.967	.999	.999
DBLP	4405	.086	1.0	.993	.999	.999
biomine	42064	.067	1.0	.996	1.0	.997

Table 3 displays some interesting and important results. Firstly, we see that for the first four graphs it is only a handful of edges that meet the requirement for applying Lyapunov CLT. However, even for a small number of edges, the Lyapunov method closely approximates DP. For *DBLP* and *biomine*, we see far more edges undergo the Lyapunov method, namely they are in the thousands and tens of thousands respectively. We observe that in all the real world graphs NUSCAN obtains near prefect matches with the clusters from USCAN. Moreover, the *RMSE* is moderately low, ranging from 0.2% to 16% depending of the dataset. In the full version of this paper we include further analysis of the parameter points.

Time Comparison. While USCAN was unable to complete tasks on our nine large graphs for most of the parameter points, it managed to finish within a reasonable time for the point (0.8, 0.5, 2) on the *enron* and *cnr-2000* graphs. However, it only completed the power law distribution for the *uk-2014-tpd* graph within the 48-hour time period. From Figure 2 we observe that NUSCAN outperforms USCAN regardless of probability distribution as the input graphs become large. Specifically for *cnr-2000* with a normal distribution of induced probabilities, NUSCAN is able to cluster the graph over 1000 times faster than USCAN. For NUSCAN is took only 16 seconds, where USCAN took 87,030 seconds (which is over a day long).

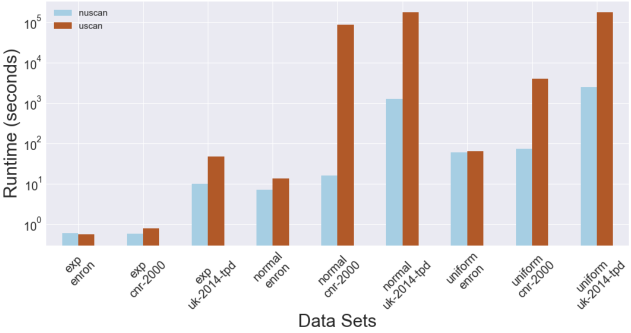

Figure 2: Running times for the three different distributions of edge probabilities. We set $\eta = 0.8$, $\varepsilon = 0.5$, $\mu = 2$, and $t = 100$. The running times for the different distributions are proportional to the number of edges that pass η pruning. For *uk-2014-tpd*, for the normal and uniform distributions, USCAN did not finish inside the 48 hours period.

In the following section, we demonstrate the scalability of NUSCAN on the nine large graphs with induced edge probabilities from the power law distribution.

4.3 Efficiency Evaluation

In this Section, we study the running time of our proposed algorithm NUSCAN over the space of parameters η, ε, μ. We set threshold $t = 100$ for each dataset. This means that we trigger the structural similarity computation using Normal Distribution only if the size of the union of neighbours for a pair of vertices is at least 100, otherwise, DP is used. Parameters η, ε, μ are varied over 55 different points in the phase space to generate a holistic sampling, and draw insights into how the parameters effect the running times.

Comparing the variation in η across the different datasets in Figure 3, the running time generally increases with the maximum degree of the dataset. As η increases, each dataset running time curve drops drastically. Figure 3 reveals that some datasets plateau off earlier than others because of the random edge probability assignment coupled with the differences in structure. For instance, with the dataset *cnr-2000*, the time drops from ten minutes when $\eta = 0.2$, all the way to one second when $\eta = 0.8$. A larger dataset such as *eswiki-2013* starts off with a time close to 30 minutes when $\eta = 0.2$, and it goes down to 25 seconds when $\eta = 0.8$. In general, all the datasets level off as η increases. Since the edge probabilities are drawn from a power law distribution, η pruning happens frequently as the value of η increases. Overall, NUSCAN completes on the largest dataset, *arabic-2005*, in less than an hour for the majority of threshold parameter points (η, ε, μ) tested. In contrast, USCAN was not able to complete in a reasonable time on any of the large datasets we tested with over 30 million edges.

Phase space. Each of the input parameters η, ε, μ have their own range of allowed values. Both η and ε are in the range $(0, 1]$, whereas $\mu \geq 2$. We can break the 55 explored points into three groups based on the fixed values $(0.2, 0.5, 2)$, $(0.5, 0.2, 2)$ and $(0.5, 0.5, 5)$. For each of the 55 runs, two of the three parameters (η, ε, μ) are held constant and the third is varied over a range of points. The range of values chosen for η and ε are $[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8]$, and μ has range $[2, 3, 4, 5, 6, 7]$. For all of the nine datasets, our algorithm completes on all points well within 48 hours; with one exception of *arabic-2005* for the three points $(0.1, \varepsilon, \mu)$ where the algorithm did not finish within the time limit.

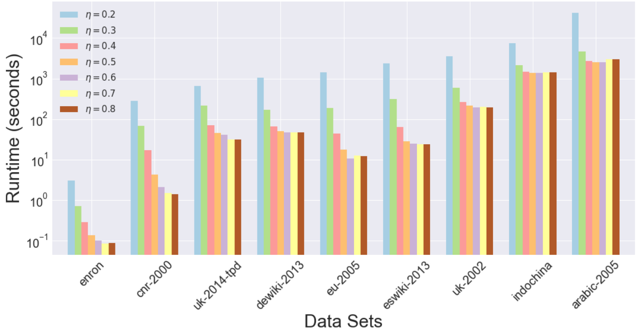

Figure 3: Running time for varying η across datasets. We set $(\eta, \varepsilon, \mu) = (\eta, 0.5, 2)$, and $t = 100$. As η increases, fewer probabilities are calculated (due to η pruning) and this reduces running time. Since the edge probabilities follow a power-law distribution, this effect plateaus as η approaches 1.

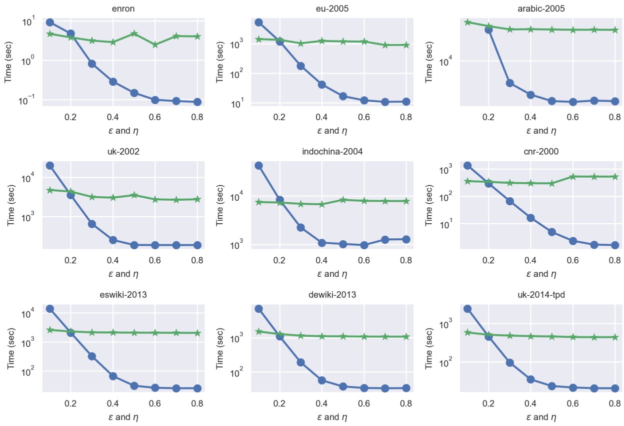

Figure 4: Running time for varying η and ε with $\mu = 2$ and $t = 100$. Since η and ε share the same range, we present both running time curves on one plot. Blue shows the variation of η, with $\varepsilon = 0.5$; green shows the variation of ε, with $\eta = 0.2$. We see that running time is mainly influenced by η.

Vanishing cluster set. Over the explored 55 points of the phase space, the number of clusters diminished as the parameters (η, ε, μ) reach the top of their ranges. The vanishing of clusters is consistent with USCAN, and thus not unique to NUSCAN. Specifically, when μ surpasses five, the number of clusters, independent of dataset, becomes zero. Since μ is the parameter responsible for determining if nodes form a cluster, the absence of clusters is bound to occur at some finite value of μ regardless of η and ε. However, clusters also vanish when η and ε become large because these parameters influence reliable structural similarity and the probability of structural similarity respectively. Consequently, when all these parameters are high, the odds of enough edges passing all the threshold requirements approaches zero. Therefore the lower half of these parameter ranges are more desirable for generating larger cluster sets.

After analyzing the running time results from each of the 55 points in the phase space, we found that only the parameter η sensibly effects the time (for more details see full version of the paper). Reflecting on the clustering algorithm, because μ is a threshold on the size of the reliable neighbourhood set to determine which vertices are reliable core vertices, then each vertex will be checked regardless of the value of μ. Similarly, ε does not effect the run time, since Algorithm 2 runs in the same time regardless of ε. However, η will effect the run time because of a pruning condition that was developed in the USCAN algorithm. Since two nodes are reliable structural similar only if $P[e, \varepsilon] \geq \eta$, then if the probability $p(e) < \eta$ that implies $P[e, \varepsilon] < \eta$, by definition of $P[e, \varepsilon]$. Hence for NUSCAN, out of the three parameters (η, ε, μ), only η significantly effects the runtime.

Figure 4 displays the effect η has on the run time. The η varying curve drops super exponentially in time over the range of chosen points, while the ε curve is a flat line. Moreover, the ε line intersects the η curve at 0.2, which is the value η is set to in Figure 4. In each of the nine plots in Figure 4, the ε line intersects the η curve right at the $\eta = 0.2$ position. Since the value of η is the parameter that dictates the running time of the process, then it is expected that the ε curve is a straight line constant in time that intersects at $\eta = 0.2$ on the η curve.

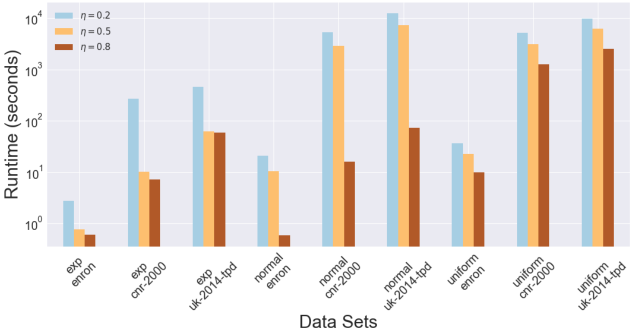

Figure 5: Running time for the three different distributions of edge probabilities. We vary η on a limited range, while setting $\varepsilon = 0.5$, $\mu = 2$, and $t = 100$. The running times for the different distributions are proportional to the number of edges that pass η pruning.

Figure 5 shows the running times for the three different probability distributions on *enron, cnr-2000*, and *uk-2014-tpd*. As η increases over its range, the running times fall off at different rates for different distributions. For the power law distribution, the time drops fast and plateaus quickly in η; for the normal distribution, the time drops off slower and falls at the end of the η range; while for the uniform distribution, the time declines at a steady pace the whole way. The reason for this variation is the η pruning that takes place. Since the power law distribution has the fewest edges with high p, it will perform the fastest. This is followed by the normal distribution, which has the majority of its edges around the mean, leading to a rapid increase in speed only at the high end of the η range. Finally, for the uniform distribution, the computation speed only marginally improves as η increases, due to the equal likelihood of edge probabilities in this distribution.

4.4 Effectiveness Testing

The only ground truth datasets known are small and already used by the authors of USCAN [35]. As we already mentioned, our clustering results are indistinguishable from those of USCAN, so there is no point repeating the same analysis as [35] for the ground truth datasets. Additionally, since our results are near identical to USCAN, we do not reproduce the comparison to other clustering algorithms done in the USCAN paper. However, what we would like to do here is to show the effectiveness of structural clustering in terms of quality for large datasets on which USCAN cannot scale, but our algorithm NUSCAN can. We start with testing a clustering metric called *Average Expected Density* (AED) defined as:

$$AED = \frac{1}{|\mathbb{C}|} \sum_{C_i \in \mathbb{C}} \sum_{e \in C_i} \frac{2p_e}{|\mathcal{V}_i| \times (|\mathcal{V}_i| - 1)} \tag{35}$$

which measures the strength of connection in each cluster averaged over all clusters, where $\mathcal{V}_i$ is the set of vertices in C_i.

In Biswas et. al. [3], they outline three metrics to measure quality of clusters when no ground truth presents itself for comparison. The authors define three metrics Average Isolability (Q_{AVI}), Average Unifiability (Q_{AVU}), and Average Isolability and Unifiability (Q_{ANUI}). Isolability determines the strength of connection within each cluster—similarly to AED—whereas Unifiability measures the strength of connection between two distinct clusters. Then Q_{ANUI} is a ratio of the average Isolability and Unifiability. For a single cluster, Isolability (I) is defined as:

$$I(C_i) = \frac{\sum\limits_{u \in_{C_i} v} p(u, v)}{\sum\limits_{u \in_{C_i} v} p(u, v) + \sum\limits_{u \in C_i, v \notin C_i} p(u, v)} \tag{36}$$

and for a pair of clusters, Unifiability (U) is defined as:

$$U(C_i, C_j) = \frac{\sum\limits_{u \in C_i, v \in C_j} p(u, v)}{\sum\limits_{u \in C_i, v \notin C_i} p(u, v) + \sum\limits_{u \notin C_j, v \in C_J} p(u, v) - \sum\limits_{u \in C_i, v \in C_j} p(u, v)} \tag{37}$$

where C_i, and C_j are different clusters in $\mathbb{C}$. Then the averages of these measures Q_{AVI} and Q_{AVU} are defined as the arithmetic mean over all cluster sets. Then Q_{ANUI} is given from the two above equations as:

$$Q_{ANUI} = \frac{Q_{AVI}}{1 + Q_{AVI} \times Q_{AVU}} \tag{38}$$

Since Q_{ANUI} is a function of both Q_{AVI} and Q_{AVU}, we display Q_{ANUI} as the objective measure.

The goal is to demonstrate that NUSCAN returns a cluster set that is as good as the USCAN cluster set under these two metrics *AED* and Q_{ANUI}. For the six smallest datasets, using the power law distribution, we are able to measure the quality of the cluster sets that NUSCAN produces. In *dewiki-2013* and *eswiki-2013*, for the ε variation curve, after $\varepsilon = 0.6$ the number of clusters found becomes zero. Hence for these two datasets, the two metrics are indeterminate for the points $(0.5, 0.7, 2)$ and $(0.5, 0.8, 2)$.

Once again we see that η is the parameter responsible for the shape of these curves. For both metrics, ε forms a flat line that intersects the η curve at 0.5, which is the value η is held constant at for the ε line.

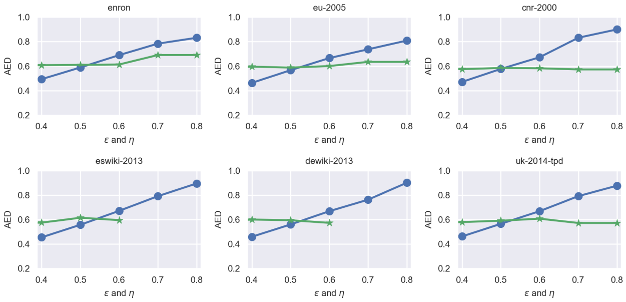

Figure 6: AED for NUSCAN when varying η and ε ($\mu = 2, t = 100$). Since η and ε share the same range of values, we show both AED curves on one plot. Blue shows the variation of η, with $\varepsilon = 0.2$ and $\mu = 2$; green shows the variation of ε, with $\eta = 0.5$ and $\mu = 2$. Again, varying ε does not influence AED much, whereas varying η has a more pronounced effect. As η increases over its ranges, AED linearly increases towards 1. Absence of some points in the ε line is due to lack of clusters at the high end of the parameter range.

The *AED* plots in Figure 6 demonstrate that as η increases from 0.4 to 0.8, the *AED* value increases from about 0.4 to 0.8-0.9 depending on the specific dataset. For instance, *enron* and *eu-2005* only make it to just over 0.8 at $\eta = 0.8$; meanwhile the four other datasets exceed 0.9 at $\eta = 0.8$. All of the η curves possess a positive linear slope. Since η is the threshold parameter that is responsible for whether two nodes are reliable structural similar, as η becomes large, exponentially less edges pass the threshold cut. Moreover, the edges that are reliably structurally similar at high η have very large edge probabilities, which leads to a high value for *AED*. As for the Q_{ANUI} metric plots in Figure 7, all the datasets quickly approach 1 in the η varied curve. Next we compare the three smallest datasets that were able to complete calculation of these metrics under the USCAN algorithm.

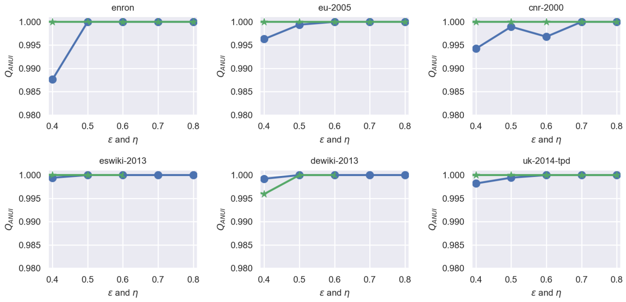

Figure 7: Q_{ANUI} for NUSCAN when varying η and ε ($\mu = 2, t = 100$) similar to Fig. 6.

Figure 8 shows both the *AED* and Q_{ANUI} plots for the three datasets *enron, cnr-2000, uk-2014-tpd*, as these were the only datasets that finished the metric calculations inside our 48 hour time constraint on USCAN. Plots of these three datasets for *AED*, and Q_{ANUI} show precisely the same curves as NUSCAN. The perfect alignment of these metric plots indicate that NUSCAN produces a near identical clustering to USCAN. Moreover, we have shown that NUSCAN does as good as USCAN on these quality metrics, meaning that with the approximation algorithm we do not sacrifice quality for the observed time improvement. Therefore, NUSCAN yields quality cluster sets that are comparable to its predecessor USCAN. In the full version we also perform this analysis on the real world graphs.

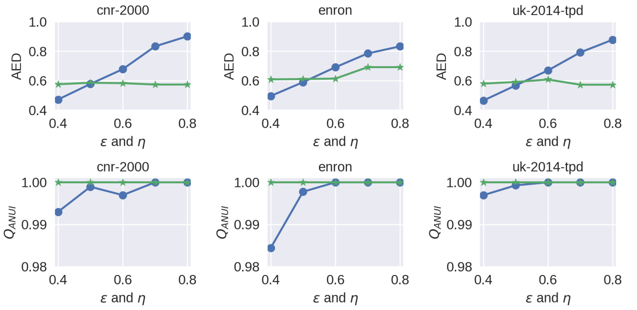

Figure 8: AED (first row) and Q_{ANUI} (second row) for USCAN when varying η and ε ($\mu = 2$) similarly to Figs. 6 and 7. Observe that AED charts are indistinguishable from the AED charts in Fig. 6 (first row of Fig. 6). Likewise, observe that Q_{ANUI} charts are also indistinguishable from the Q_{ANUI} charts in Fig. 7 (first row of Fig. 7). So, our approximation method, NUSCAN, produces results virtually same as USCAN in practice.

5 RELATED WORK

Since the original publication of the SCAN paper, many improvements and additions have been built on top of SCAN. Some authors [10, 40, 46] insert parallel processing paradigms to improve practical performance of calculating the structural similarity. The paper by Che et. al. [9] develops a min-max pruning method to improve the performance on detecting core vertices. For Seo et. al. [38] they demonstrate how detecting and merging local clusters scales the performance of the clustering framework. In Chang et. al. [7] they prove that the SCAN algorithm is worst case optimal and introduce new scaleable techniques that practically improve the structural clustering procedure. The structural similarity formula is an integral component of the SCAN framework and many authors have explored variations of this ratio to improve performance. Recently, papers have been adopting the Jaccard similarity as the equation for structural similarity [7, 35, 37].

In Qiu et. al. [35] the authors derive new definitions that construct a probabilistic structural clustering algorithm, called USCAN. The key idea of USCAN is the notion of probability of structural similarity, which calculates the probability that a pair of vertices connected by an edge have a structural similarity (Jaccard similarity) above ε. Hence rather than having $\varepsilon-$neighbourhoods, where nodes in the set have structural similarities larger than ε; there are $(\varepsilon, \eta)-$reliable neighbourhoods, where vertices in the set have a probability greater than η that the edge pair has a structural similarity greater than ε over all possible worlds. Then using the reliable neighbourhoods, reliable core vertices are determined and the algorithm from there follows the remainder of the SCAN protocol.

The novelty of USCAN is the definition of the probability of structural similarity, which the authors show can be calculated in polynomial time by dynamically iterating over classes of neighbourhoods for each edge. However, the Dynamic Program that calculates the probability is the bottleneck of the entire program, taking quadratic time with respect to the union of neighbourhoods between the edge. As a result of the time complexity, USCAN can not scale to large graph datasets with over 30 million edges.

Liang et. al. [28] claim to improve the time complexity of USCAN with a new formulation of the calculation for the probability of structural similarity. Their algorithm called ProbSCAN, displayed an equation for the bottleneck process that calculates the probability of structural similarity in linear time with respect to the union of neighbourhoods, rather than the quadratic time in USCAN. The paper does not give a proof, and unfortunately, we are able to show using a counterexample that ProbSCAN produces incorrect results (see Appendix in the full version of the paper).[1] Hence, we regard USCAN as the current state-of-art algorithm for structural clustering of probabilistic graphs.

There are also other clustering frameworks that apply to probabilistic graphs. Some methods extend the framework of K-Means clustering to probabilistic graphs by maximizing the average connection probability in each of the k clusters [6, 19]. With Halim et. al. [18] the approach for clustering probabilistic graphs analyzes the surrounding neighbourhood of vertices for an edge and calculates a weighted average to decide whether the edge in question passes a static threshold cut. The authors of [13] utilize a variation of Jaccard similarity that uses random walks to identify similarity, then feeds the probabilities into an encoder and deep learning network with a Gaussian embedding to discover the resulting clusters. Other methods for clustering non-graph data include a density-based algorithm (DBSCAN) which identifies dense regions in the data [17, 44, 45]. There are also algorithms that use sampling to find clusters in probabilistic graphs [21, 31]. Our approach is distinct from these works in terms of problem definition and techniques used. We focus on structural clustering which classifies the vertices into three types of nodes, namely, cores, hubs, and outliers, and uses a metric to determine the type of each vertex in the network. From the techniques point of view, we are the first to utilize the Lyapunov CLT for clustering probabilistic graphs. Other works have used it to solve problems unrelated to clustering [12, 14–16]. Previous works used the technique in the context of computing probabilistic graph decompositions such as k-core and k-truss. For [15, 16] it was straightforward to show that the conditions necessary to apply the Lyapunov CLT held; it is much more mathematically demanding in this work to show that the conditions for Lyapunov CLT hold due to the nature of the random variable V we need to consider for our problem and requires new technical insights.

6 CONCLUSIONS

Clustering probabilistic networks is a complex and time consuming endeavor with low to moderate scalability. The proposed algorithm NUSCAN offers an approximation solution to the probabilistic clustering problem. By applying a novel Lyapunov CLT approximation to the calculation for the probability of structural similarity, the entire clustering process takes $O\left(\alpha \left[m_D t^2 + m_L d_{max}\right]\right)$ time in the worst case. When compared to USCAN, our approximation algorithm produces near identical clusters with up to three orders of magnitude time improvement on some datasets.

[1] We informed the authors of the error through personal communication.

REFERENCES

[1] Charu C Aggarwal. 2013. *Managing and mining sensor data.* Springer Science & Business Media.

[2] Joel S Bader, Amitabha Chaudhuri, Jonathan M Rothberg, and John Chant. 2004. Gaining confidence in high-throughput protein interaction networks. *Nature biotechnology* 22, 1 (2004), 78–85.

[3] Anupam Biswas and Bhaskar Biswas. 2017. Defining quality metrics for graph clustering evaluation. *Expert Systems with Applications* 71 (2017), 1–17.

[4] Paolo Boldi, Francesco Bonchi, Aris Gionis, and Tamir Tassa. 2012. Injecting uncertainty in graphs for identity obfuscation. *arXiv preprint arXiv:1208.4145* (2012).

[5] Francesco Bonchi, Francesco Gullo, Andreas Kaltenbrunner, and Yana Volkovich. 2014. Core decomposition of uncertain graphs. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.* 1316–1325.

[6] Matteo Ceccarello, Carlo Fantozzi, Andrea Pietracaprina, Geppino Pucci, and Fabio Vandin. 2017. Clustering uncertain graphs. *Proceedings of the VLDB Endowment* 11, 4 (2017), 472–484.

[7] Lijun Chang, Wei Li, Lu Qin, Wenjie Zhang, and Shiyu Yang. 2017. pSCAN: Fast and Exact Structural Graph Clustering. *IEEE Transactions on Knowledge and Data Engineering* 29, 2 (2017), 387–401.

[8] Sudarshan S Chawathe. 2019. Clustering blockchain data. In *Clustering Methods for Big Data Analytics.* Springer, 43–72.

[9] Yulin Che, Shixuan Sun, and Qiong Luo. 2018. Parallelizing pruning-based graph structural clustering. In *Proceedings of the 47th International Conference on Parallel Processing.* 1–10.

[10] Jia-Jun Chen, Ji-Meng Chen, Jie Liu, and Va-Lou Huang. 2013. PSCAN: A parallel structural clustering algorithm for networks. In *2013 International Conference on Machine Learning and Cybernetics*, Vol. 2. IEEE, 839–844.

[11] Norishige Chiba and Takao Nishizeki. 1985. Arboricity and subgraph listing algorithms. *SIAM Journal on computing* 14, 1 (1985), 210–223.

[12] Alfredo Cuzzocrea, Edoardo Fadda, and Alessandro Baldo. 2021. Lyapunov Central Limit Theorem: Theoretical Properties and Applications in Big-Data-Populated Smart City Settings. In *2021 5th International Conference on Cloud and Big Data Computing (ICCBDC).* 34–38.

[13] Malihe Danesh, Morteza Dorrigiv, and Farzin Yaghmaee. 2022. DGCU: A new deep directed method based on Gaussian embedding for clustering uncertain graphs. *Computers and Electrical Engineering* 101 (2022), 108066.

[14] Can Ding, Jing Zhang, Yingjie Zhang, Zhe Zhang, and Xiaoyao Li. 2021. Neural Network-Based ADP Cotrol for Nonliear Systems with Prescribed Performance Constriant. In *2021 33rd Chinese Control and Decision Conference (CCDC).* IEEE, 6342–6346.

[15] Fatemeh Esfahani, Mahsa Daneshmand, Venkatesh Srinivasan, Alex Thomo, and Kui Wu. 2022. Scalable probabilistic truss decomposition using central limit theorem and H-index. *Distributed and Parallel Databases* 40, 2 (2022), 299–333.

[16] Fatemeh Esfahani, Venkatesh Srinivasan, Alex Thomo, and Kui Wu. 2019. Efficient Computation of Probabilistic Core Decomposition at Web-Scale.. In *EDBT.* 325–336.

[17] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *kdd*, Vol. 96. 226–231.

[18] Zahid Halim, Muhammad Waqas, Abdul Rauf Baig, and Ahmar Rashid. 2017. Efficient clustering of large uncertain graphs using neighborhood information. *International Journal of Approximate Reasoning* 90 (2017), 274–291.

[19] Kai Han, Fei Gui, Xiaokui Xiao, Jing Tang, Yuntian He, Zongmai Cao, and He Huang. 2019. Efficient and effective algorithms for clustering uncertain graphs. *Proceedings of the VLDB Endowment* 12, 6 (2019), 667–680.

[20] Petteri Hintsanen. 2007. The most reliable subgraph problem. In *PKDD*, Vol. 2007. Springer, 471–478.

[21] Syed Fawad Hussain, Ifra Arif Butt, Muhammad Hanif, and Sajid Anwar. 2022. Clustering uncertain graphs using ant colony optimization (ACO). *Neural Computing and Applications* (2022), 1–18.

[22] Ruoming Jin, Lin Liu, Bolin Ding, and Haixun Wang. 2011. Distance-constraint reachability computation in uncertain graphs. *Proceedings of the VLDB Endowment* 4, 9 (2011), 551–562.

[23] Haruko Kawahigashi, Yoshiaki Terashima, Naoto Miyauchi, and Tetsuo Nakakawaji. 2005. Modeling ad hoc sensor networks using random graph theory. In *Second IEEE Consumer Communications and Networking Conference, 2005. CCNC. 2005.* IEEE, 104–109.

[24] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining.* 137–146.

[25] Yun Joong Kim, Kiyong Kim, Heonwoo Lee, Junbeom Jeon, Jinwoo Lee, and Jeehee Yoon. 2022. The Protein-Protein Interaction Network of Hereditary Parkinsonism Genes Is a Hierarchical Scale-Free Network. *Yonsei medical journal* 63, 8 (2022), 724.

[26] Nevan J Krogan, Gerard Cagney, Haiyuan Yu, Gouqing Zhong, Xinghua Guo, Alexandr Ignatchenko, Joyce Li, Shuye Pu, Nira Datta, Aaron P Tikuisis, et al. 2006. Global landscape of protein complexes in the yeast Saccharomyces cerevisiae. *Nature* 440, 7084 (2006), 637–643.

[27] Ugur Kuter and Jennifer Golbeck. 2007. Sunny: A new algorithm for trust inference in social networks using probabilistic confidence models. In *AAAI*, Vol. 7. 1377–1382.

[28] Yongjiang Liang, Tingting Hu, and Peixiang Zhao. 2020. Efficient Structural Clustering in Large Uncertain Graphs. In *2020 IEEE 36th International Conference on Data Engineering (ICDE).* IEEE, 1966–1969.

[29] Chenhao Ma, Reynold Cheng, Laks VS Lakshmanan, Tobias Grubenmann, Yixiang Fang, and Xiaodong Li. 2019. Linc: a motif counting algorithm for uncertain graphs. *Proceedings of the VLDB Endowment* 13, 2 (2019), 155–168.

[30] Ian McCulloh, Joshua Lospinoso, and KM Carley. 2007. Social network probability mechanics. In *Proceedings of the World Scientific Engineering Academy and Society 12 th International Conference on Applied Mathematics.*

[31] Naoto Ohsaka, Tomohiro Sonobe, Sumio Fujita, and Ken-ichi Kawarabayashi. 2017. Coarsening massive influence networks for scalable diffusion analysis. In *Proceedings of the 2017 ACM International Conference on Management of Data.* 635–650.

[32] David Ong and Jue Wang. 2015. Income attraction: An online dating field experiment. *Journal of Economic Behavior & Organization* 111 (2015), 13–22.

[33] Symeon Papadopoulos, Ioannis Kompatsiaris, Athena Vakali, and Ploutarchos Spyridonos. 2012. Community detection in Social Media. *Data Min. Knowl. Discov.* 24 (05 2012), 515–554. https://doi.org/10.1007/s10618-011-0224-z

[34] Michalis Potamias, Francesco Bonchi, Aristides Gionis, and George Kollios. 2010. K-nearest neighbors in uncertain graphs. *Proceedings of the VLDB Endowment* 3, 1-2 (2010), 997–1008.

[35] Yu-Xuan Qiu, Rong-Hua Li, Jianxin Li, Shaojie Qiao, Guoren Wang, Jeffrey Xu Yu, and Rui Mao. 2018. Efficient structural clustering on probabilistic graphs. *IEEE Transactions on Knowledge and Data Engineering* 31, 10 (2018), 1954–1968.

[36] Panyu Ren, Xiaodi Yang, Tianpeng Wang, Yunpeng Hou, and Ziding Zhang. 2022. Proteome-wide prediction and analysis of the Cryptosporidium parvum protein-protein interaction network through integrative methods. *Computational and Structural Biotechnology Journal* (2022).

[37] Boyu Ruan, Junhao Gan, Hao Wu, and Anthony Wirth. 2021. Dynamic structural clustering on graphs. In *Proceedings of the 2021 International Conference on Management of Data.* 1491–1503.

[38] Jung Hyuk Seo and Myoung Ho Kim. 2017. pm-SCAN: an I/O efficient structural clustering algorithm for large-scale graphs. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management.* 2295–2298.

[39] Brian Skyrms and Robin Pemantle. 2009. A dynamic model of social network formation. In *Adaptive networks.* Springer, 231–251.

[40] Thomas Ryan Stovall, Sinan Kockara, and Recep Avci. 2014. GPUSCAN: GPU-based parallel structural clustering algorithm for networks. *IEEE Transactions on Parallel and Distributed Systems* 26, 12 (2014), 3381–3393.

[41] Peng Xia, Kun Tu, Bruno Ribeiro, Hua Jiang, Xiaodong Wang, Cindy Chen, Benyuan Liu, and Don Towsley. 2014. Characterization of user online dating behavior and preference on a large online dating site. In *Social Network Analysis-Community Detection and Evolution.* Springer, 193–217.

[42] Xiaowei Xu, Nurcan Yuruk, Zhidan Feng, and Thomas AJ Schweiger. 2007. Scan: a structural clustering algorithm for networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining.* 824–833.

[43] Samuel Zahl. 1966. Bounds for the central limit theorem error. *SIAM J. Appl. Math.* 14, 6 (1966), 1225–1245.

[44] Xianchao Zhang, Han Liu, and Xiaotong Zhang. 2017. Novel density-based and hierarchical density-based clustering algorithms for uncertain data. *Neural networks* 93 (2017), 240–255.

[45] Xianchao Zhang, Han Liu, Xiaotong Zhang, and Xinyue Liu. 2014. Novel density-based clustering algorithms for uncertain data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 28.

[46] Weizhong Zhao, Venkataswamy Martha, and Xiaowei Xu. 2013. PSCAN: a parallel Structural clustering algorithm for big networks in MapReduce. In *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA).* IEEE, 862–869.

[47] Zhaonian Zou, Hong Gao, and Jianzhong Li. 2010. Discovering frequent subgraphs over uncertain graph databases under probabilistic semantics. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining.* 633–642.

[48] Zhaonian Zou, Jianzhong Li, Hong Gao, and Shuo Zhang. 2010. Finding top-k maximal cliques in an uncertain graph. In *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010).* IEEE, 649–652.