Xihui Chen, Sjouke Mauw, and Yunior Ramírez-Cruz*

Publishing Community-Preserving Attributed Social Graphs with a Differential Privacy Guarantee

Abstract: We present a novel method for publishing differentially private synthetic attributed graphs. Our method allows, for the first time, to publish synthetic graphs simultaneously preserving structural properties, user attributes and the community structure of the original graph. Our proposal relies on CAGM, a new community-preserving generative model for attributed graphs. We equip CAGM with efficient methods for attributed graph sampling and parameter estimation. For the latter, we introduce differentially private computation methods, which allow us to release community-preserving synthetic attributed social graphs with a strong formal privacy guarantee. Through comprehensive experiments, we show that our new model outperforms its most relevant counterparts in synthesising differentially private attributed social graphs that preserve the community structure of the original graph, as well as degree sequences and clustering coefficients.

Keywords: attributed social graphs, generative models, differential privacy, community detection

DOI Editor to enter DOI
Received ..; revised ..; accepted ...

1 Introduction

The use of online social networks (OSNs) has grown steadily during the last years, and is expected to continue growing in the future. The ubiquity of OSNs has turned them into one of the most important sources of data for the analysis of social phenomena. Such analyses have led to significant findings used in a wide range of applications, from efficient epidemic disease control [8, 36] to information diffusion [21, 66]. Despite the social benefits that can be obtained from social network analysis, access to social data by third parties such as researchers and companies must be limited due to the sensitivity of the information stored in OSNs, e.g. personal relationships, political preferences and religious affiliations. In addition, the increase of public awareness about privacy and the entry into effect of strong privacy regulations such as the GDPR [1] strengthen the reluctance of OSN owners to release part of the data they hold. Therefore, it is of critical importance to provide mechanisms for privacy-preserving social data publication to encourage OSN owners to release data for analysis and to provide strong privacy guarantees to their users.

Social graphs are a natural representation of social networks, with nodes corresponding to participants and edges to connections between participants. A large number of methods have been devised for publishing sanitised versions of the original social graph [5, 7, 9, 10, 33–35, 38, 39, 50, 52, 53, 58, 65, 67], computing graph statistics in a privacy-preserving manner [15, 24, 63], or releasing synthetic graphs that preserve properties of the original graph while protecting user's private information [16, 42, 51, 54, 59]. Among these methods, *differential privacy* (DP) [13] has gained enormous popularity due to its strong privacy guarantees and the fact that it is a semantic privacy notion, focusing on the data processing algorithms rather than specific datasets or types of adversary knowledge. According to the type of published data, we can divide differentially private mechanisms for social graphs into two classes. The methods in the first class directly release specific statistics of the social graph, e.g. the degree sequence [15, 24] or the number of specific subgraphs (triangles, stars, etc.) [63]. The second family of methods focuses on publishing synthetic social graphs as a replacement of real social networks in a two-step process [42, 51, 54, 59]. In the first step, differentially private methods are used to compute the parameters of a generative model that accurately captures the original graph properties. Then, in the second step, synthetic graphs are sampled from the private model. DP requires to define in advance a *privacy budget*,

Xihui Chen: SnT, University of Luxembourg, Esch-sur-Alzette, Luxembourg, E-mail: xihui.chen@uni.lu
Sjouke Mauw: DCS, University of Luxembourg, Esch-sur-Alzette, Luxembourg, E-mail: sjouke.mauw@uni.lu
***Corresponding Author: Yunior Ramírez-Cruz:** SnT, University of Luxembourg, Esch-sur-Alzette, Luxembourg, E-mail: yunior.ramirez@uni.lu

which determines the amount of perturbation that will be applied to the outputs of algorithms. In consequence, methods in the first family need to either limit in advance the number of queries to answer or deliver increasingly lower quality answers. On the contrary, methods in the second family can devote the entire privacy budget to the model parameter estimation, without further degradation of the sampled graphs. For this reason, in this paper we focus on the second type of methods.

For analysts, the utility of synthetic graphs is determined by the ability of the graph models to capture relevant properties of the original graph. To satisfy this need, several graph models have been proposed to accurately capture global structural properties such as degree distributions and clustering coefficients, as well as heterogeneous user attributes such as gender, education or marital status. However, the best differentially private models proposed so far fail to capture the community structure of the original graph. Informally, a community is a set of users who are much more interrelated among themselves than to other users of the network. Examples of communities are: a group of Gmail users who frequently e-mail each other, or a group of employees of the same company. The emergence of communities has been shown to be an inherent property of social networks [49, 61]. Analysts would tremendously benefit from the availability of synthetic attributed graphs that preserve the community structure of the original graph. For example, they may be able to improve online shopping recommendations based on the common purchases of users belonging to the same community. Current models and methods are insufficient for enabling such an analysis, as they either lack information about the community structure or they lack vertex features.

In this paper, we address the problem discussed in the previous paragraph by introducing, for the first time, a generative graph model that simultaneously preserves global network properties, user attributes and community structure. Our model is called CAGM (*Community-Preserving Attributed Graph Model*). It is equipped with efficient parameter estimation and graph sampling methods. Furthermore, we provide differentially private variants of the CAGM's parameter estimation methods, which allow us to release synthetic attributed social graphs with a strong privacy guarantee and increased utility with respect to preceding approaches.

Summary of contributions:

- We introduce CAGM, the first generative attributed graph model that simultaneously captures a number of properties of the community structure, along with user attributes and structural properties.
- We present efficient methods for learning an instance of our model from an input graph and sampling community-preserving synthetic attributed graphs from this instance. We show, via a number of experiments on real-world social networks, that the community structures of synthetic graphs sampled from our model are more similar to those of the original graphs than those of the graphs sampled from previous models. Additionally, we show that this behaviour is obtained without sacrificing the ability to preserve global structural features.
- We devise differentially private methods for computing the parameters of the new model. We empirically demonstrate that differentially private synthetic attributed graphs generated by our model suffer a reasonably low degradation with respect to their counterparts, in terms of their ability to capture the community structure and structural features of the original graphs.

2 Related Work

Private graph synthesis. The key to synthesising social graphs is the model which determines both the information embedded in the published graphs and the properties preserved. Mir et al. [42] used the Kronecker graph generative model [30] to generate differentially private graphs. As the Kronecker model cannot accurately capture structural properties, Sala et al. [51] proposed an alternative approach which makes use of the dK-graph model, which is based on counting the occurrences of specific subgraphs with K vertices (e.g. length-K paths). Wang et al. [54] further improved the work of Sala et al. by considering global sensitivity instead of local sensitivity (refer to Sect. 3 for the definition of sensitivity). Xiao et al. [59] used the *hierarchical random graph* (HRG) model [12] and found that it can further reduce the amount of added noise and thus increase the accuracy. Recently, Zhang et al. [64] proposed an edge plausibility measure to help decide which edges seem more natural when added to a graph as a part of an anonymisation method. They applied this measure as an extension to Sala et al.'s method, in particular to its implementation included in the SecGraph library [19]. Zhang et al. pointed out that this implementation mostly adds fake edges to the original graph, rather than synthesising a new graph from scratch.

The approaches described so far work on unlabelled graphs. Pfiffer et al. [18] introduced the *attributed graph*

model (AGM) which attaches binary attributes to nodes and captures the correlations between shared attributes and the existence of connections. Jorgensen et al. [20] adopted this model and proposed differentially private methods to accurately estimate the model parameters. They also designed a new graph generation method based on the *transitive Chung-Lu* (TCL) model [17], which enables the model to sample attributed graphs preserving the clustering coefficient. As discussed previously, CAGM, the model introduced in this paper, is comparable to Jorgensen et al.'s model in preserving global structural properties of the original graph, but it outperforms it by also capturing the community structure.

Private statistics publishing. Degree sequences and degree correlations are two types of the statistics frequently studied in the literature. The general trend in publishing these statistics under DP consists in adding noise to the original sequences and then post-processing the perturbed sequences to enforce or restore certain properties, such as graphicality [24], vertex order in terms of degrees [15], etc. Subgraph count queries, e.g. the number of triangles or k-stars, have also received considerable attention. Among the approaches to accurately compute such queries, we have ladder functions [63] and smooth sensitivity [23, 55].

The aforementioned approaches have focused on unlabelled (non-attributed) graphs. The task of publishing graph statistics from attributed social networks was addressed in [32]. This work focuses on studying the effect of dependences between tuples on differentially private methods for computing the degree sequence of a graph. They consider node attributes to be public information, and use the overlap between node attributes as a model of dependences between nodes. Then, they apply the stronger notion of dependent differential privacy (DDP), also introduced in the paper, to compute degree sequences in the presence of an adversary who knows the true values of node attributes. Our work differs from the one presented in [32] in the fact that we do not treat node features as public information, but as sensitive information, and apply DP for estimating attribute distributions. Regarding the applicability of DDP in our approach, a case where it may be applicable is that of an adversary who can extract tuple dependencies from additional public knowledge, e.g. location traces which are obtainable from check-ins in social network posts. While being an interesting direction for future work, such an extension goes beyond the scope of this work, since it would require new DDP mechanisms for estimating all the parameters of our model under every possible scenario in terms of adversary knowledge.

Community-preserving graph generation models. A number of existing random graph models claim to capture community structure, e.g., *block two-level Erdős-Rényi* (BTER) [26], ILFR [49], *stochastic block model* (SBM) [56] and its variants (e.g., DCSBM [22] and DCPPM [45]), *attributed networks with communities generator* (ANCG) [29] and DANCer [28]. BTER generates community-preserving social graphs given expected node degrees and, for every degree value σ, the average of the clustering coefficients of the nodes of degree σ. The model assumes that every community is a set of σ nodes with degree σ. On the contrary, CAGM makes no assumptions on the community partition received. Moreover, ILFR and the variants of SBM preserve edge densities at the community level but, unlike our new model, they do not preserve the clustering coefficients of the original graph. Finally, ANCG and DANCer aim to synthesise graphs satisfying some known behaviours of complex networks such as small world, preferential attachment and homophily, extending the behaviour of the classic Barabási-Albert model [2]. Although these models take node attributes and community structure into account, they only provide generation methods given user-defined parameters. Since their goals do not include learning model parameters from existing graphs, they do not provide estimation methods for model parameters and, consequently, privacy preservation mechanisms are not necessary for their application scenario.

Community-enhanced de-anonymisation and community-preserving anonymisation. A number of works on user de-anonymisation have shown that knowledge about communities can be exploited by an adversary to improve the re-identification of pseudonymised users [14, 47, 57]. The anonymisation scenario in which these attacks are applied differs fundamentally from the one described in this paper. Our approach views node attributes and edges as private, but considers vertex ids to be public. As a consequence, synthetic graphs generated by our mechanism share the same vertex set of the original graph. Our purpose is not to hide users' identities. Therefore, in this scenario, de-anonymisation is not a threat. Previous works reported in the literature additionally support our rationale that the goals of graph anonymisation and community structure preservation are not necessarily in contradiction. For example, edge edition operations that better preserve community structures of the original graph during anonymisation are studied in [6], whereas a community-preserving anonymisation method is proposed in [50]. A similar behaviour is shown by synthetic graphs generated by our approach, when

node ids are pseudonymised. Interested readers can refer to App. E, where we empirically show that pseudonymised versions of our synthetic graphs also resist the strongest community-enhanced de-anonymisation attack proposed in the literature [47].

3 Preliminaries

3.1 Notations

An attributed graph is represented as a triple $G = (\mathcal{V}, \mathcal{E}, X)$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, and X is a binary matrix called the *attribute matrix*. The i-th row of X is the attribute vector of v_i, which is individually denoted by $\tau(v_i)$. Every column of X represents a binary feature, which is set to 1 (**true**), or 0 (**false**), for each user. For example, if the j-th column represents the attribute "*owning a car*", $X_{ij} = 1$ means that the user represented by v_i owns a car. Non-binary real-life attributes are assumed to be binarised. The order of the columns of X is fixed, but arbitrary, and has no impact on the results described hereafter. Throughout the paper, we deal with undirected graphs. That is, if $(v_i, v_j) \in \mathcal{E}$, then $(v_j, v_i) \in \mathcal{E}$. Additionally, we use A to denote the adjacency matrix of the graph.

We use $\mathcal{C} = \{C_0, C_1, \ldots, C_p\}$, with $C_i \subseteq \mathcal{V}$ for every $i \in \{0, 1, \ldots, p\}$, to represent a *community partition* of the attributed graph. As the term suggests, in this paper we assume that $C_i \cap C_j = \emptyset$, with $0 \leq i < j \leq p$, and $\cup_{C_i \in \mathcal{C}} C_i = \mathcal{V}$. The community C_0 has a special interpretation. Since some community detection algorithms assign no community to some vertices, we will use C_0 as a "discard" community of unassigned vertices. We do so to avoid having a potentially large number of singleton communities, for which no meaningful co-affiliation statistics can be computed. We use $\psi_{\mathcal{C}}(v_i)$ to denote the community to which the node v_i belongs in the community partition $\mathcal{C}$. If $\mathcal{C}$ is clear from the context, we will simply use $\psi(v_i)$. Table 1 summarises the most important notation used throughout the paper.

Table 1. Important notations.

$G = (\mathcal{V}, \mathcal{E}, X)$	An attributed graph G
$\mathcal{V}$	Set of nodes of G
$\mathcal{E}$	Set of edges of G
X	Attribute matrix of G
$\tau(v)$	Attribute vector of $v \in \mathcal{V}$
$\mathcal{C}$	Community partition of G
$\psi_{\mathcal{C}}(v)$	Community to which v belongs in $\mathcal{C}$
$\beta(\tau(v), \tau(w))$	Descriptor of the edge (v, w) in terms of $\tau(v)$ and $\tau(w)$
$\mathcal{M} = \langle \mathcal{V}, \mathcal{C}, \Theta_M^c, \Theta_X^c, \Theta_F^c \rangle$	An instance of CAGM
Θ_M^c	Edge generation model
Θ_X^c	Attribute vector generation model
Θ_F^c	Attribute-edge correlations generation model
$d_{intra}(v)$	Intra-community degree of v
$d_{inter}(v)$	Inter-community degree of v
$n_{\triangle}^{intra}$	Number of intra-community triangles
$n_{\triangle}^{inter}$	Number of inter-community triangles
n_C^{ℓ}	Number of nodes in $C \in \mathcal{C}$ whose ℓ-th attribute has value 1
r_C^u	Number of intra-community edges (v, w) in $C \in \mathcal{C}$ such that $\beta(\tau(v), \tau(w)) = u$
r_{inter}^u	Number of inter-community edges (v, w) such that $\beta(\tau(v), \tau(w)) = u$

3.2 Differential Privacy

Differential privacy [13] is a well studied statistical notion of privacy. The intuition behind it is to randomise the output of an algorithm in such a way that the presence of any individual element in the input dataset has a negligible impact on the probability of observing any particular output. In other words, a mechanism is ε-differentially private if for any pair of *neighbouring datasets*, i.e. datasets that only differ by one element, the probabilities of obtaining any output are measurably similar. The amount of similarity is determined by the parameter ε, which is commonly called the *privacy budget*. In what follows, we will use the notation $\mathcal{D}$ for the set of possible datasets, $\mathcal{O}$ for the set of possible outputs, and $D \sim D'$ for a pair of neighbouring datasets.

Definition 1 (ε-differential privacy [13])**.** *A randomised mechanism $\mathcal{M}\colon \mathcal{D} \to \mathcal{O}$ satisfies ε-differential privacy if for every pair of neighbouring datasets $D, D' \in \mathcal{D}$, $D \sim D'$, and for every $S \subseteq \mathcal{O}$, we have* $\Pr(\mathcal{M}(D) \in S) \leq e^{\varepsilon} \Pr(\mathcal{M}(D') \in S)$.

A number of differentially private mechanisms have been proposed. For queries of the form $q\colon \mathcal{D} \to \mathbb{R}^n$, the most widely used mechanism to enforce differential privacy is the *Laplace mechanism*, which consists in obtaining the (non-private) output of q and adding to every component a carefully chosen amount of random noise, which is drawn from the Laplace distribution $Lap(\lambda)\colon f(y \mid \lambda) = \frac{1}{2\lambda} \exp(\frac{-|y|}{\lambda})$, where y is a real-valued variable indicating the noise to be added, $\lambda = \frac{\Delta_q}{\varepsilon}$ and Δ_q is a property of the original function q called *global sensitivity*. This property is defined as the largest diffe-

rence between the outputs of q for any pair of neighbouring datasets, that is $\Delta_q = \max_{D \sim D'} \|q(D) - q(D')\|_1$, where $\|\cdot\|_1$ is the L_1 norm. For categorical queries of the form $q\colon \mathcal{D} \to \mathcal{O}$, where $\mathcal{O}$ is a finite set of categories, the *exponential mechanism* [41] is the most commonly used. In this case, for each value $o \in \mathcal{O}$, a score is assigned by a function (usually called *scoring function*) quantifying the value's utility, denoted by $u(o, D)$. The global sensitivity of u is $\Delta_u = \max_{o \in \mathcal{O}, D \sim D'} |u(o, D) - u(o, D')|$, and the randomised output is drawn with probability proportional to $\exp(\frac{\varepsilon \cdot u(o,D)}{2\Delta_u})$. Differentially private methods are composable [40], and deterministic post-processing of the output of an ε-differentially private algorithm also satisfies ε-differential privacy [25]. These properties allow us to divide a complex computation, such as the set of model parameters, into a sequence of sub-tasks for which differentially private methods exist or can be more easily developed.

4 The CAGM Model

In this section we give the formal definition of CAGM. We introduce the methods for sampling synthetic graphs from the model, and describe the methods for learning the model parameters from an attributed graph.

4.1 Overview

Alg. 1 summarises the process by which CAGM is used for publishing synthetic attributed graphs. A thorough description of the parameters of CAGM is given in Sect. 4.2, and parameter estimation is discussed in Sect. 4.4. Once the model parameters have been estimated, we can sample any number of synthetic attributed graphs from the model, as described in steps 3 to 7 of Alg. 1. Thus, the synthetic graphs generated by Alg. 1 have the same vertex set as the original graph, whereas the attribute matrix and the edge set are sampled from the model (step 6). For every new synthetic attributed graph, we first sample the attribute matrix, and then this matrix is used, in combination with an edge generation model (Sect. 4.3.1), to generate the edge set of the synthetic graph. There are two reasons for dividing this process into two steps. The first one is to make the sampling process efficient. The second reason is to profit from the two-step process to enforce the intuition that users with features of certain patterns are more likely to be connected in the social network. The attributed graph sampling procedure is discussed in detail in Sect. 4.3. To conclude, note that if steps 1 and 2 are executed in a differentially private manner, the synthetic graphs obtained by steps 3 to 7 also satisfy DP, because the generation is done as a post-processing step of the differentially private computation.

Algorithm 1: Given $G = (\mathcal{V}, \mathcal{E}, X)$, obtain t attributed synthetic graphs.

1 Obtain community partition;
2 Estimate CAGM parameters;
3 **for** $i \in \{1, 2, \ldots, t\}$ **do**
4 | Sample X_i from CAGM;
5 | Sample $\mathcal{E}_i$ from CAGM;
6 | $G_i \leftarrow (\mathcal{V}, \mathcal{E}_i, X_i)$

4.2 Model Parameters

As we discussed in Sect. 1, given an attributed graph G and a community partition $\mathcal{C}$ of G, the purpose of CAGM is to capture a number of properties of $\mathcal{C}$ that are overlooked by previously defined models, without sacrificing the ability to capture global structural properties such as degree distributions and clustering coefficients. To that end, CAGM models the following properties of the community partition:

1. the number and sizes of communities;
2. the number of intra-community edges in every community;
3. the number of inter-community edges;
4. the distributions of attribute vectors in every community;
5. the distributions of the so-called *attribute-edge correlations* [20], for the set of inter-community edges and for the set of intra-community edges in every community.

Graphs generated by CAGM will have the same number of vertices as the original graph, as well as the same number of communities. Moreover, every community will have the same cardinality as in the original graph, and the same number of intra-community edges. The number of inter-community edges of the generated graph will also be the same as that of the original graph. Notice that the model preserves the total number, but not necessarily the pairwise numbers of inter-community edges for every pair of communities.

Attribute-edge correlations were defined in [20] as heuristic values for characterising the relation between the feature vectors labelling a pair of vertices and the likelihood that these vertices are connected. They encode

the intuition that, for example, co-workers who attended the same university and live near to each other are more likely to be friends than persons with fewer features in common, whereas friends are more likely to support the same sports teams or go to the same bars than unrelated persons. In our model, we compute one distribution of attribute-edge correlations for each community, as well as an inter-community distribution.

A key element in the representation of attribute-edge correlations is the notion of *aggregator functions*. An aggregator function $\beta\colon \{0,1\}^k \times \{0,1\}^k \to \mathcal{B}$ maps a pair of attribute vectors x, x' of dimensionality k into a value in a discrete range $\mathcal{B}$, which is used as a descriptor, also called *aggregated feature*, of the pair (x, x'). For example, $\mathcal{B}$ can contain a set of similarity levels for pairs of feature vectors, such as $\{$*low*, *medium*, *high*$\}$, and β can map a pair of vectors whose cosine similarity is in the interval $[0, 0.33]$ to *low*, a pair of vectors whose cosine similarity is in the interval $[0.67, 1]$ to *high*, etc. Attribute-edge correlations, along with the community-wise distributions of attribute vectors, are useful for analysts, as they allow to characterise the members of a community in terms of frequently shared features, hypothesise explanations for the emergence of a community, etc.

Formally, an instance of the CAGM model is defined as a quintuple $\mathcal{M} = \langle \mathcal{V}, \mathcal{C}, \Theta_M^c, \Theta_X^c, \Theta_F^c \rangle$, where:

- $\mathcal{V}$ is a set of vertices.
- $\mathcal{C}$ is a community partition of $\mathcal{V}$.
- Θ_M^c is an instance of an edge set generative model that preserves properties 1 to 3 of the community partition $\mathcal{C}$, as well as degree distributions and clustering coefficients.
- Θ_X^c is an instance of an attribute vector generation model, which aims to preserve property 4 of the community partition. The model defines, for every attribute vector x, every $C \in \mathcal{C}$ and every $v \in C$, the probability $\Pr(\tau(v) = x \,|\, C, \Theta_X^c)$ that a vertex in C is labelled with x.
- Θ_F^c is an instance of a generative model for attribute-edge correlations, which aims to preserve property 5 of the community partition. This model defines:
  - The discrete range $\mathcal{B}$ and aggregator function β.
  - The probability
    $$\Pr(\beta(\tau(v_i), \tau(v_j)) = u \,|\, \psi_{\mathcal{C}}(v_i) = \psi_{\mathcal{C}}(v_j) = C, A_{i,j} = 1, \Theta_F^c)$$
    for every community $C \in \mathcal{C}$ and every $u \in \mathcal{B}$.
  - The probability
    $$\Pr(\beta(\tau(v_i), \tau(v_j)) = u \,|\, \psi_{\mathcal{C}}(v_i) \neq \psi_{\mathcal{C}}(v_j), A_{i,j} = 1, \Theta_F^c)$$
    for every $u \in \mathcal{B}$.

4.3 Sampling Attributed Graphs from an Instance of CAGM

Given an instance $\mathcal{G} = \langle \mathcal{V}, \mathcal{C}, \Theta_M^c, \Theta_X^c, \Theta_F^c \rangle$ of CAGM, with $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$, an attributed graph $G = (\mathcal{V}, \mathcal{E}, X)$ is sampled from $\mathcal{G}$ with probability $\Pr(G \mid \mathcal{G}) = \Pr(\mathcal{E}, X \mid \mathcal{G})$ which, for tractability, is approximated as

$$\Pr(\mathcal{E}, X \,|\, \mathcal{C}, \Theta_M^c, \Theta_F^c, \Theta_X^c) = \Pr(\mathcal{E} \,|\, X, \mathcal{C}, \Theta_M^c, \Theta_F^c) \cdot \Pr(X \,|\, \mathcal{C}, \Theta_X^c).$$

That is, we first sample from Θ_X^c the attribute vectors labelling each vertex and then use them in sampling the edge set. Again, for tractability, we make

$$\Pr(X \,|\, \mathcal{C}, \Theta_X^c) = \prod_{v_i \in \mathcal{V}} \Pr(\tau(v_i) = x_i \,|\, \psi_{\mathcal{C}}(v_i), \Theta_X^c),$$

where x_i is the i-th row of X. Probabilities of the form $\Pr(\tau(v_i) = x_i \,|\, \psi_{\mathcal{C}}(v_i), \Theta_X^c)$ are estimated from the input graph, as will be described in Sect. 4.4. Then, under the assumption that edges are sampled independently, we approximate $\Pr(\mathcal{E} \mid X, \mathcal{C}, \Theta_M^c, \Theta_F^c)$ in terms of attribute-edge correlations as follows:

$$\Pr(\mathcal{E} \,|X, \mathcal{C}, \Theta_M^c, \Theta_F^c) = \prod_{v_i, v_j \in \mathcal{V}} \Pr(A_{i,j} = 1 \,|\, \beta(\tau(v_i), \tau(v_j)), \mathcal{C}, \Theta_M^c, \Theta_F^c).$$

An efficient two-step procedure was presented in [18] for sampling edges from the distribution $\Pr(A_{i,j} = 1 \mid \beta(\tau(v_i), \tau(v_j)), \Theta_M^c, \Theta_F^c)$. Here, we adapt this procedure for sampling edges from the distribution $\Pr(A_{i,j} = 1 \mid \Theta_M^c, \Theta_F^c, \beta(\tau(v_i), \tau(v_j)), \mathcal{C})$. In the first step, a candidate edge (v_i, v_j) is sampled from the edge generation model Θ_M^c (which does not account for node features) with probability $Q'_M(i,j) = \frac{\Pr(A_{i,j}=1 | \mathcal{C}, \Theta_M^c)}{\sum_{v_p, v_q \in \mathcal{V}} \Pr(A_{p,q}=1 | \mathcal{C}, \Theta_M^c)}$, where the probabilities of the form $\Pr(A_{i,j} = 1 \mid \mathcal{C}, \Theta_M^c)$ are estimated from the input graph. In the second step, the candidate edge is accepted as an edge of G with probability $\Gamma(\beta(\tau(v_i), \tau(v_j)), \mathcal{C})$, which is computed in two different manners depending on the community co-affiliation of v_i and v_j. If $\psi_{\mathcal{C}}(v_i) = \psi_{\mathcal{C}}(v_j) = C$, we first compute

$$R_{intra}(\beta(\tau(v_i), \tau(v_j)), C) = \frac{\Pr(\beta(\tau(v_i), \tau(v_j)) | \psi_{\mathcal{C}}(v_i) = \psi_{\mathcal{C}}(v_j) = C, A_{i,j} = 1, \Theta_F^c)}{\Pr_M(\beta(\tau(v_i), \tau(v_j)) | \psi_{\mathcal{C}}(v_i) = \psi_{\mathcal{C}}(v_j) = C, A_{i,j} = 1, \Theta_M^c, \Theta_F^c)},$$

which represents the ratio between the probability that an edge joining two nodes in community C of the input graph is described by $\beta(\tau(v_i), \tau(v_j))$ and the probability that an edge generated by Θ_M^c, and joining two nodes in C, is described by $\beta(\tau(v_i), \tau(v_j))$. Then, we make

$$\Gamma(\beta(\tau(v_i), \tau(v_j)), \mathcal{C}) = \Gamma_{intra}(\beta(\tau(v_i), \tau(v_j)), C) = \frac{R_{intra}(\beta(\tau(v_i), \tau(v_j)), C)}{SupR}.$$

Algorithm 2: SampleFromCAGM($\mathcal{V}, \mathcal{C}, \Theta_M^c, \Theta_X^c, \Theta_F^c$)

$X' \leftarrow$ SampleAttributeVectors(Θ_X^c);
$Q'_M \leftarrow$ ComputeQM($\Theta_M^c, \mathcal{C}$);
$\mathcal{E}' \leftarrow$ SampleEdgeSet(Q'_M);
for $s \in \mathcal{B}$ **do**
  Compute $\Gamma_{inter}(s)$;
  for $C \in \mathcal{C}$ **do**
    Compute $\Gamma_{intra}(s, C)$
$\mathcal{E}' \leftarrow \emptyset$;
while $|\mathcal{E}'| < |\mathcal{E}|$ **do**
  $(v, w) \leftarrow$ SampleEdge(Q'_M);
  $s \leftarrow \beta(\tau(v), \tau(w))$;
  $u \leftarrow$ Uniform$(0, 1)$;
  if $(\psi_{\mathcal{C}}(v) = \psi_{\mathcal{C}}(w) \wedge u \leq \Gamma_{intra}(s, \psi_{\mathcal{C}}(v))$ ***or*** $(\psi_{\mathcal{C}}(v) \neq \psi_{\mathcal{C}}(w) \wedge u \leq \Gamma_{inter}(s)$ **then**
    $\mathcal{E}' \leftarrow \mathcal{E}' \cup \{(v, w)\}$;
return $X', \mathcal{E}'$;

On the contrary, if $\psi_{\mathcal{C}}(v_i) \neq \psi_{\mathcal{C}}(v_j)$, we first compute

$$R_{inter}(\beta(\tau(v_i), \tau(v_j))) = \frac{\Pr(\beta(\tau(v_i), \tau(v_j)) | \psi_{\mathcal{C}}(v_i) \neq \psi_{\mathcal{C}}(v_j), A_{i,j}=1, \Theta_F^c)}{\Pr_M(\beta(\tau(v_i), \tau(v_j)) | \psi_{\mathcal{C}}(v_i) \neq \psi_{\mathcal{C}}(v_j), A_{i,j}=1, \Theta_M^c, \Theta_F^c)},$$

which represents the ratio between the probability that an edge joining two nodes from different communities in the input graph is described by $\beta(\tau(v_i), \tau(v_j))$ and the probability that an edge generated by Θ_M^c, and joining two nodes from different communities, is described by $\beta(\tau(v_i), \tau(v_j))$. Then, we make

$$\Gamma(\beta(\tau(v_i), \tau(v_j)), \mathcal{C}) = \Gamma_{inter}(\beta(\tau(v_i), \tau(v_j))) = \\ = \frac{R_{inter}(\beta(\tau(v_i), \tau(v_j)))}{SupR}.$$

In computing the values of both Γ_{intra} and Γ_{inter},

$$SupR = \sup \cup_{s \in \mathcal{B}, C \in \mathcal{C}} \left(R_{intra}(s, C) \cup R_{inter}(s) \right).$$

Alg. 2 describes the procedure to sample an attributed graph from CAGM.

4.3.1 Edge generation model

As we discussed in Sect. 4.2, the component Θ_M^c of CAGM is an edge generation model which preserves several properties of the community partition of the original graph (properties 1 to 3 listed in Sect. 4.2), in addition to the degree distribution and clustering coefficients. We call this model *Community-preserving Graph Model* (CPGM). It is an extension of the TriCycle model, introduced in [20], that takes community structure into account. CPGM takes as input the set of vertices, as well as the expected number of neighbours of every vertex v within its community (that is, its *intra-community degree*, denoted by $d_{intra}(v)$) and the expected number of neighbours outside its community (that is, the *inter-community degree*, denoted by $d_{inter}(v)$). These values are used to enforce the expected densities within every community and between communities. Additionally, the model also requires the number of triangles having all vertices in one community (which we call *intra-community triangles* and denote by $n_{\triangle}^{intra}$), as well as the number of triangles spanning more than one community (*inter-community triangles*, denoted by $n_{\triangle}^{inter}$). It was shown in [20] that synthetic graphs that preserve the number of triangles of the original graph are more likely to approximate its clustering coefficient. Moreover, both $n_{\triangle}^{intra}$ and $n_{\triangle}^{inter}$ can be efficiently and accurately computed under DP. In CPGM, the edge generation process consists of two steps. The first step produces a graph that preserves the intra- and inter-community degrees, but not the number of intra- and inter-community triangles. Then, the second step iteratively edits the original edge set until $n_{\triangle}^{intra}$ and $n_{\triangle}^{inter}$ are approximately enforced, within a tolerance threshold. The second step is based on the observation presented in [17] that the clustering behaviour in social networks stems from the higher likelihood of users with common friends to connect to each other, thus creating triangles.

At the first step, we follow the idea of the CL model [11]. For every pair of vertices v and w satisfying $\psi_{\mathcal{C}}(v) = \psi_{\mathcal{C}}(w) = C$, the intra-community edge (v, w) is added with probability $\pi_C^{intra}(v, w) = \frac{d_{intra}(v) d_{intra}(w)}{2m_C^{intra}}$, where m_C^{intra} is the original number of intra-community edges in C. That is, intra-community edges are added with a probability proportional to the product of the intra-community degrees of the linked vertices. If $\psi_{\mathcal{C}}(v) \neq \psi_{\mathcal{C}}(w)$, then the inter-community edge (v, w) is added with probability $\pi^{inter}(v, w) = \frac{d_{inter}(v) d_{inter}(w)}{2m^{inter}}$, where m^{inter} is the total number of inter-community edges in the original graph. Alg. 3 describes the first step of the generation process. The second step of the generation process consists in iteratively applying edge swaps until the numbers of intra- and inter-community triangles are approximately enforced, within a 98% tolerance window. Every edge swap consists in first adding a new edge and then removing another. If the added edge is an intra-community edge, then the oldest intra-community edge (in terms of the order of creation by Alg. 3) is removed, even if it does not belong to the same community. Likewise, if the added edge is an inter-community edge, then the oldest inter-community edge is removed. In the method, all intra-community edge swaps are executed

Algorithm 3: GenInitialEdgeSet($d_{intra}, d_{inter}, \mathcal{C}$)

$\mathcal{E} \leftarrow \emptyset$;
for $C \in \mathcal{C}$ **do**
  $m_C^{intra} \leftarrow \frac{1}{2}\sum_{v \in C} d_{intra}(v)$;
  $m \leftarrow 0$;
  while $m \leq m_C^{intra}$ **do**
    $(v, w) \leftarrow \mathsf{Sample}(\pi_C^{intra})$;
    if $(v, w) \notin \mathcal{E}$ **then**
      $\mathcal{E} \leftarrow \mathcal{E} \cup \{(v, w)\}$;
      $m \leftarrow m + 1$;
$m^{inter} \leftarrow \frac{1}{2}\sum_{v \in \mathcal{V}} d_{inter}(v)$;
while $m \leq \sum_{C \in \mathcal{C}} m_C^{intra} + m^{inter}$ **do**
  $(v, w) \leftarrow \mathsf{Sample}(\pi^{inter})$;
  if $(v, w) \notin \mathcal{E}$ **then**
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{(v, w)\}$;
    $m \leftarrow m + 1$;

Algorithm 4: GetFinalEdgeSet($d_{intra}, d_{inter}, n_\triangle^{intra}, n_\triangle^{inter}, \mathcal{C}$)

$\mu_\triangle^{intra} \leftarrow \mathsf{CountIntraCommTriangles}(\mathcal{E})$;
while $\mu_\triangle^{intra} < n_\triangle^{intra}$ **do**
  Uniformly sample C from $\mathcal{C}$;
  Sample v_1 from C with probability $\frac{d_{intra}(v_1)}{2m_C^{intra}}$;
  Uniformly sample v_2 from $\mathcal{N}_{intra}(v_1)$;
  Uniformly sample v_3 from $\mathcal{N}_{intra}(v_2)$;
  if $(v_1, v_3) \notin \mathcal{E} \wedge v_3 \neq v_1$ **then**
    $(v_1', v_2') \leftarrow \mathsf{GetOldestIntraCommEdge}(\mathcal{E}, \mathcal{C})$;
    $n_{cn}^{prev} \leftarrow \mathsf{GetCommonNeighbour}(v_1', v_2')$;
    $\mathcal{E} \leftarrow \mathcal{E} / \{(v_1', v_2')\}$;
    $n_{cn}^{new} \leftarrow \mathsf{GetCommonNeighbour}(v_1, v_3)$;
    if $n_{cn}^{prev} < n_{cn}^{new}$ **then**
      $\mathcal{E} \leftarrow \mathcal{E} \cup \{(v_1, v_3)\}$;
      $\mu_\triangle^{intra} \leftarrow \mu_\triangle^{intra} - n_{cn}^{prev} + n_{cn}^{new}$;
    else
      $\mathcal{E} \leftarrow \mathcal{E} \cup \{(v_1', v_2')\}$;
$\mu_\triangle^{inter} \leftarrow \mathsf{CountInterCommTriangles}(\mathcal{E})$;
while $\mu_\triangle^{inter} < n_\triangle^{inter}$ **do**
  Sample v_1 from $\mathcal{V}$ with probability $\frac{d_{inter}(v_1)}{2m^{inter}}$;
  Uniformly sample v_2 from $\mathcal{N}_{inter}(v_1)$;
  Uniformly sample v_3 from $\mathcal{N}_{intra}(v_2)$;
  $(v_1', v_2') \leftarrow \mathsf{GetOldestInterCommEdge}(\mathcal{E}, \mathcal{C})$;
  $n_{cn}^{prev} \leftarrow \mathsf{GetCommonNeighbour}(v_1', v_2')$;
  $\mathcal{E} \leftarrow \mathcal{E} / \{(v_1', v_2')\}$;
  $n_{cn}^{new} \leftarrow \mathsf{GetCommonNeighbour}(v_1, v_3)$;
  if $n_{cn}^{prev} < n_{cn}^{new}$ **then**
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{(v_1, v_3)\}$
    $\mu_\triangle^{inter} \leftarrow \mu_\triangle^{inter} - n_{cn}^{prev} + n_{cn}^{new}$;
  else
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{(v_1', v_2')\}$;

before inter-community edge swaps. The reason for this is that adding or removing an intra-community edge may change the number of inter-community triangles as well, whereas inter-community triangles can be created without modifying the number of intra-community triangles. Edge swaps resulting in a reduction in the number of intra- or inter-community triangles are discarded, in which case the edge selected for removal is set to be the youngest in the graph so it is not selected again in a sufficiently large number of coming iterations. To reduce the likelihood of discarding edge swaps, the edge selected for addition must be one that transforms at least one *wedge* (a length-3 path not forming a triangle) into an intra- or inter-community triangle, as required. The edge edition method is described in Alg. 4. In the algorithm, we denote by $\mathcal{N}_{intra}(v)$ the set of neighbours of v in its community, that is $\mathcal{N}_{intra}(v) = \{w \mid \psi_{\mathcal{C}}(v) = \psi_{\mathcal{C}}(w) \wedge (v, w) \in \mathcal{E}\}$. Likewise, we denote by $\mathcal{N}_{inter}(v)$ the set of neighbours of v in different communities, that is $\mathcal{N}_{inter}(v) = \{w \mid \psi_{\mathcal{C}}(v) \neq \psi_{\mathcal{C}}(w) \wedge (v, w) \in \mathcal{E}\}$.

Due to the removal of initially generated edges, the synthetic graph may become disconnected. In this case, we apply an edge-swapping post-processing step to reconnect every small connected component to the main component (the connected component with the most nodes). If the post-processing reduces the number of triangles, we recall Alg. 4. The alternation between the post-processing and Alg. 4 is not guaranteed to yield a graph having exactly the required number of triangles, so we stop the iteration when the total number of triangles in the synthetic graph is within a 98% tolerance window with respect to the original one. We also note that Alg. 4 is not guaranteed to enforce $n_\triangle^{intra}$ and $n_\triangle^{inter}$ in all cases. This issue is inherited from TriCycle, but it does not occur in realistic social graphs as the ones used in our experiments, neither in those used in [20].

4.4 Parameter Estimation for CAGM

Estimating Θ_M^c. The estimation of Θ_M^c reduces to computing the community-wise counters that it relies on: intra- and inter-community degrees of every vertex, the number of intra-community triangles for each community and the number of inter-community triangles. As we mentioned in Sect. 4.3.1, degrees and triangle counts are used to preserve global structural properties of the generated graphs such as degree distribution and clustering coefficients. They can be efficiently computed on the original graph both exactly and under DP.

Estimating Θ_X^c. In order to keep the estimation procedure tractable, we introduce the assumption that attributes are independent. While not realistic in all scenarios, this assumption simplifies the estimation and handles the sparsity of the attribute vectors when the num-

ber of attributes is large. As seen in [18, 20], not having such an assumption severely limits the number of features that can be practically handled. We will denote by x_ℓ be the value for the ℓ-th component of the attribute of vector x. Likewise, we will denote by $\tau_\ell(v)$ the value of the ℓ-th component of the vector labelling vertex v. We estimate the probability that a node v is labelled with an attribute vector x by the following formula:

$$\Pr(\tau(v) = x \,|\, \psi_{\mathcal{C}}(v), \Theta_X^c) = \prod_{\ell=1}^{k} \Pr_\ell(\tau_\ell(v) = x_\ell \mid \psi_{\mathcal{C}}(v), \Theta_X^c),$$

where k is the number of columns of X (ergo the cardinality of all attribute vectors) and

$$\Pr_\ell(\tau_\ell(v) = x_\ell \mid \psi_{\mathcal{C}}(v), \Theta_X^c) = \tfrac{|\{v' \in \psi_{\mathcal{C}}(v) | \tau_\ell(v') = x_\ell\}|}{|\psi_{\mathcal{C}}(v)|}.$$

Estimating Θ_F^c. As we discussed in Sect. 4.2, for defining Θ_F^c it is necessary to define an aggregator function for pairs of attribute vectors. Our aggregator function is based on the widely used cosine similarity, that is, the cosine of the angle between the two feature vectors. Since the range $\mathcal{B}$ of aggregator functions needs to be discrete, we split the range $[0, 1]$ of the cosine similarity into a set of intervals, determined by a parameter δ satisfying $0 < \delta \leq 1$. Let $s_{cos}(x, x')$ denote the similarity between vectors x and x'. Our aggregator function is defined as $\beta(x, x') = \left\lfloor \frac{s_{cos}(x,x')}{\delta} \right\rfloor$. Note that, according to this definition, $\mathcal{B} = \{\lfloor \frac{s}{\delta} \rfloor \mid s \in [0, 1]\}$. Finally, the probability that an aggregated feature $u \in \mathcal{B}$ characterises a pair of connected vertices v_i, v_j satisfying $\psi_{\mathcal{C}}(v_i) = \psi_{\mathcal{C}}(v_j) = C$ in the input graph is computed as

$$\Pr(\beta(\tau(v_i), \tau(v_j)) = u \,|\, \psi_{\mathcal{C}}(v_i) = \psi_{\mathcal{C}}(v_j) = C, A_{i,j} = 1, \Theta_F^c) = \frac{|\{(v_p, v_q) \in \mathcal{E} \mid \beta(\tau(v_p), \tau(v_q)) = u \wedge \psi(v_p) = \psi(v_q) = C\}|}{|\{(v_p, v_q) \in \mathcal{E} \mid \psi(v_p) = \psi(v_q) = C\}|}.$$

Analogously, for a pair of connected vertices v_i, v_j satisfying $\psi_{\mathcal{C}}(v_i) \neq \psi_{\mathcal{C}}(v_j)$, we make

$$\Pr(\beta(\tau(v_i), \tau(v_j)) = u \,|\, \psi_{\mathcal{C}}(v_i) \neq \psi_{\mathcal{C}}(v_j), A_{i,j} = 1, \Theta_F^c) = \frac{|\,\{(v_p, v_q) \in \mathcal{E} \mid \beta(\tau(v_p), \tau(v_q)) = u \wedge \psi(v_p) \neq \psi(v_q)\}\,|}{|\,\{(v_p, v_q) \in \mathcal{E} \mid \psi(v_p) \neq \psi(v_q)\,|}.$$

Compared to the approach introduced in [18, 20], our model uses a coarser granularity for aggregated features. Thanks to that, it avoids computing 2^{2k} different probability values, which is extremely inefficient.

5 Differentially Private CAGM

As we discussed in Sect. 3, the difference between instantiations of DP for graphs lies in the definition of the graph pairs considered to be neighbouring datasets. Here, we adopt the following definition from [20].

Definition 2 (Neighbouring attributed graphs [20])**.** *A pair of attributed graphs $G = (\mathcal{V}, \mathcal{E}, X)$ and $G' = (\mathcal{V}, \mathcal{E}', \mathcal{X}')$ are* neighbouring*, denoted $G \sim_{at} G'$, if and only if they differ in the presence of exactly one edge or the attribute vector of exactly one node. That is,*

$$G \sim_{at} G' \iff |\mathcal{E} \nabla \mathcal{E}'| = 1 \ \vee (\exists_{v \in \mathcal{V}} \ \tau_G(v) \neq \tau_{G'}(v) \wedge \wedge \forall_{v' \in \mathcal{V} \setminus \{v\}} \ \tau_G(v) = \tau_{G'}(v)).$$

Def. 2 entails that the existence of relations, that is the occurrence of edges, and the attributes describing every particular user, are treated as sensitive. On the contrary, vertex identifiers are treated as non-private. These criteria are in line with the current privacy policies of most social networking sites, where the fact that a profile exists is public information, but users can keep their personal information and friends list private or hidden from the general public. With Def. 2 in mind, we describe in what follows the differentially private computation of every parameter of CAGM. Due to space limitations, we develop all the proofs in the appendices.

5.1 Obtaining the Community Partition

Our differentially private community partition method extends the algorithm ModDivisive [46], in such a way that it takes node attributes into account. ModDivisive searches for a community partition that maximises *modularity*, a structural parameter encoding the intuition that a user tends to be more connected to users in the same community than to users in other communities [44]. Modularity is defined as $\sum_{C \in \mathcal{C}} \left(\frac{\ell_C}{m} - \left(\frac{d_C}{2m} \right)^2 \right)$, where ℓ_C is the number of edges between the nodes in C and d_C is the sum of degrees of the nodes in C. ModDivisive uses the exponential mechanism, considering the set of possible partitions as the categorical co-domain, and using modularity as the scoring function. In order to integrate node features into ModDivisive, we introduce a new objective function that combines the original modularity with an attribute-based quality criterion. The new function is defined as $Q(\mathcal{C}) = w_s \cdot Q_s(\mathcal{C}) + w_a \cdot Q_a(\mathcal{C})$, where $w_s \in [0, 1]$, $w_a = 1 - w_s$, $Q_s(\mathcal{C})$ is the modularity of the original graph and $Q_a(\mathcal{C})$ is the modularity of an auxiliary graph obtained from the original as follows. First, we take the vertex set of the original graph. Then, we compute all pairwise similarities between their associated feature vectors. Similarities are computed using the cosine measure Finally, we add to the auxiliary graph the edges corresponding to the $\left\lceil \frac{n(n-1)}{20} \right\rceil$ most similar attributed node pairs, that is the top-ranked 10%. It is proven in [46] that the global sensitivity of $Q_s(\mathcal{C})$ is

upper bounded by $\frac{3}{m}$, where m is the minimum number of edges of all potential graphs to publish. In the worst case, $\Delta_{Q_s(\mathcal{C})} = 3$, considering that the original graph is an arbitrary non-empty graph. However, this is not the case for real-life social graphs, so introducing more realistic assumptions about the value of m allows us to use smaller values of $\Delta_{Q_s(\mathcal{C})}$ and thus reduce the amount of noise added in differentially privately computing $Q_s(\mathcal{C})$. Throughout this paper, we assume $m = 10,000$, which leads to $\Delta_{Q_s(\mathcal{C})} = 0.0003$. In what follows, we apply an analogous reasoning for bounding $\Delta_{Q_a(\mathcal{C})}$.

Proposition 1. *Every graph G of order n satisfies $LS_{Q_a(\mathcal{C})}(G) \leq \frac{60}{n}$.*

Combining the result in [46] with that of Prop. 1, we conclude that $LS_{Q(\mathcal{C})}(G) \leq 0.0003 \cdot w_s + \frac{60}{|\mathcal{V}(G)|} \cdot w_a$ for every G satisfying the aforementioned assumptions, and use this value as an upper bound for $\Delta_{Q_a(\mathcal{C})}$.

5.2 Attribute Vector Distribution

As discussed in Sect. 4.4, given a community partition $\mathcal{C}$, in order to obtain the differentially private estimation of Θ_X^c (denoted by $\overline{\Theta}_X^c$), we need to compute the probability distribution of each attribute for every community, i.e. $\Pr_\ell(\tau_\ell(v) = x_\ell \mid \psi_{\mathcal{C}}(v), \overline{\Theta}_X^c)$, for each $\ell \in \{1, \ldots, k\}$ and $C \in \mathcal{C}$ (recall that k is the number of attributes). Computing this probability reduces to computing the number of nodes in C whose ℓ-th attribute has value 1, which we denote n_C^ℓ. Let n_C be the k-uple $\left(n_C^1, n_C^2, \ldots, n_C^k\right)$. In order to obtain the differentially private k-uple $\overline{n}_C = \left(\overline{n}_C^1, \overline{n}_C^2, \ldots, \overline{n}_C^k\right)$, we add to each element in n_C noise sampled from $Lap\left(\frac{k}{\varepsilon_X}\right)$, where ε_X is the privacy budget reserved for this computation and k is the global sensitivity of n_C, as shown next.

Proposition 2. *The global sensitivity of n_C is k.*

5.3 Attribute-Edge Correlations

Recall that the aggregator function β defined in Sect. 4.4 maps every pair of attribute vectors to a non-negative integer in $\mathcal{B} = \{\lfloor \frac{s}{\delta} \rfloor \mid s \in [0,1]\}$, for some $\delta \in [0,1]$. In order to estimate Θ_F^c, we need to count, for each possible output of β, the number of edges whose endnodes are mapped to this value. For every $u \in \mathcal{B}$ and every $C \in \mathcal{C}$, let r_C^u be the number of intra-community edges (v,w) in C such that $\beta(\tau(v), \tau(w)) = u$. Likewise, let r_{inter}^u be the number of inter-community edges (v,w) such that $\beta(\tau(v), \tau(w)) = u$. Thus, in order to compute $\overline{\Theta}_F^c$, we need to differentially privately compute r_C^u for every $u \in \mathcal{B}$ and every $C \in \mathcal{C}$, as well as r_{inter}^u for every $u \in \mathcal{B}$. We denote by $\overline{r}_C^u$ and $\overline{r}_{inter}^u$ the corresponding differentially private values.

The global sensitivity of every $|\mathcal{B}|$-uple of the form $\left(\overline{r}_C^{u_1}, \overline{r}_C^{u_2}, \ldots, \overline{r}_C^{u_{|\mathcal{B}|}}\right)$, $C \in \mathcal{C}$, as well as that of every $|\mathcal{B}|$-uple of the form $\left(\overline{r}_{inter}^{u_1}, \overline{r}_{inter}^{u_2}, \ldots, \overline{r}_{inter}^{u_{|\mathcal{B}|}}\right)$, is $2(|\mathcal{V}| - 2)$ [20], which is unbounded. To overcome this problem, we follow an approach analogous to the one used in [20] for counting attribute-edge correlations in the entire graph. The method, introduced in [3], consists in truncating the edge set of the graph to ensure that the degree of all nodes is at most p, which in the case of attribute-edge correlations ensures that the global sensitivity is $2p$ [3, 20]. In consequence, for every $u \in \mathcal{B}$ and every $C \in \mathcal{C}$, we obtain $\overline{r}_C^u$ from r_C^u by adding noise sampled from $Lap(\frac{2p}{\varepsilon_F})$, where where ε_F is the privacy budget reserved for this computation. Likewise, we obtain $\overline{r}_{inter}^u$ from r_{inter}^u by adding noise sampled from $Lap(\frac{2p}{\varepsilon_F})$.

5.4 CPGM Parameters

Intra- and inter-community degrees. We first add noise to the raw degree values and then apply a post-processing on the perturbed degree sequences to restore certain properties of the original sequence, namely graphicality and the order of the nodes in terms of their degrees, as well as certain community-specific properties. Let $d_{intra}^C = (d_{intra}^{1,C}, d_{intra}^{2,C} \ldots, d_{intra}^{m,C})$, where $m = |C|$ and $d_{intra}^{i,C} \leq d_{intra}^{i+1,C}$ $(1 \leq i < |C|)$, be the list of non-decreasingly ordered original intra-community degrees in $C \in \mathcal{C}$. Analogously, let $d_{inter}^C = (d_{inter}^{1,C}, d_{inter}^{2,C}, \ldots, d_{inter}^{m,C})$ be the sequence of inter-community degrees of nodes in C. The global sensitivity of the degree sequence of the entire graph is 2, as adding or removing one edge changes the degrees of exactly two nodes by 1 [15]. The same is true for every d_{intra}^C and d_{inter}^C, since the degrees of at most two intra-community nodes (or at most one node in C and one node outside of C) change by 1. Thus, for every $C \in \overline{\mathcal{C}}$, we obtain from d_{intra}^C the differentially private sequence $\overline{d}_{intra}^C$ by adding noise sampled from $Lap(\frac{2}{\varepsilon_d})$ to every degree value. Similarly, we obtain from d_{inter}^C the differentially private sequence $\overline{d}_{inter}^C$. Afterwards, the noisy sequences are post-processed to restore three properties: (i) the non-decreasing order between the intra-community degrees inside every community, (ii) the graphicality of the intra-community degrees of every community, and (iii) the graphicality of the inter-community degrees of all nodes in the graph. Property (i) is enforced using the method proposed in [15],

whereas properties (ii) and (iii) are enforced using the method proposed in [24].

Numbers of intra- and inter-community triangles. The global sensitivity of the number of triangles in a graph is proven in [48] to be $n-2$, where n is the number of vertices. The next result characterises the global sensitivity of the number of intra-community triangles.

Proposition 3. *The global sensitivity of the number of intra-community triangles of a graph G with a community partition $\mathcal{C}$ is $\Delta_{n_\triangle^{intra}} = \max_{C\in\mathcal{C}}\{|C|-2\}$.*

Since the global sensitivity of triangle count queries is unbounded, the Laplace mechanism cannot be applied in this case. An accurate differentially private method for counting the number of triangles of a graph is presented in [63]. This method uses the exponential mechanism. It interprets the triangle count query as a categorical query, whose co-domain is a partition $\mathcal{O}$ of $\mathbb{Z}^+$. One of the elements of $\mathcal{O}$ is a singleton set composed exclusively of the correct output of the query, whereas every other element contains a set of inaccurate values which are treated as equally useful. They define the notion of *ladder function*, which is used as a scoring function on the elements of $\mathcal{O}$. A ladder function gives better scores to the sets of values that are closer to the correct query answer. In order to differentially privately compute the number of triangles of a graph G, it is first necessary to compute the correct number of triangles. Then, the ladder function is built, and a set $O \in \mathcal{O}$ is sampled following the exponential mechanism. Finally, a random element of O is given as the differentially private output of the query [63]. It is shown in [63] that the best ladder function, in the sense that it adds the minimum necessary amount of noise, is the so-called *local sensitivity at distance t* [48], denoted as $LS_q(G,t)$, which is based on the notion of *local sensitivity*. The local sensitivity of a query q on a graph G [48] is computed as $LS_q(G) = \max_{G\sim G'} \|q(G)-q(G')\|_1$, that is the maximum difference between the output of q on G and those on its neighbouring datasets. The local sensitivity at distance t is defined as the maximum local sensitivity of the query q among all the graphs at edge-edit distance at most t from G. Formally, $LS_q(G,t) = \max_{\{G' \mid \phi(G,G')\leq t\}} LS_q(G')$, where $\phi(G,G')$ is the edge-edit distance between G and G'. It is also shown in [48, 63] that $LS_q(G,t) = \max_{1\leq i<j\leq|\mathcal{V}|} LS^q_{ij}(G,t)$, where $LS^q_{ij}(G,t) = \max_{\{G',G'' \mid \phi(G,G')\leq t, G'\sim_{ij}G''} |q(G')-q(G'')|$ and $G' \sim_{ij} G''$ indicates that G' and G'' differ in exactly the addition or removal of (v_i,v_j).

Here, we apply the ladder function approach for computing the number of intra-community triangles. To that end, we characterise the function $LS_{n_\triangle^{intra}}((G,\mathcal{C}),t)$ for every graph G with a community partition $\mathcal{C}$.

Proposition 4. *For every graph G, every community partition $\mathcal{C}$ of G, and every positive integer $t \geq 1$,*

$$LS_{n_\triangle^{intra}}((G,\mathcal{C}),t) = \max_{\{i,j \mid \psi_\mathcal{C}(v_i)=\psi_\mathcal{C}(v_j)\}} \left\{ \min \left\{ a_{ij} + \left\lfloor \frac{t+\min\{t,b_{ij}\}}{2} \right\rfloor, |\psi_\mathcal{C}(v_i)|-2 \right\} \right\},$$

where $a_{ij} = |\{v_\ell \in \psi_\mathcal{C}(v_i) \mid A_{i,\ell} = 1 \wedge A_{j,\ell} = 1\}|$ and $b_{ij} = |\{v_\ell \in \psi_\mathcal{C}(v_i) \mid A_{i,\ell} \oplus A_{j,\ell} = 1\}|$.

In Prop. 4, the operator $\oplus$ denotes exclusive or. Notice that $LS_{n_\triangle^{intra}}((G,\mathcal{C}),t)$ can be efficiently computed for small values of t, and it converges to the efficiently computable global sensitivity $\Delta_{n_\triangle^{intra}}$ for $t \geq 2\max_{C\in\mathcal{C}}|C|$, so it can be used for efficiently and privately computing the number of intra-community triangles. Finally, for computing the number of inter-community triangles, we use the method from [63] to compute the number of triangles of the entire graph, and subtract from it the number of intra-community triangles computed with the method described above.

5.5 Summary and Discussion

In what follows, we will use the notation CAGMDP to refer to a differentially private instance of CAGM. The privacy budget ε is split among the different computations as follows: $\varepsilon_c = \frac{\varepsilon}{2}$ for the community partition method, $\varepsilon_F = \frac{\varepsilon}{6}$ for the estimation of $\overline{\Theta}^c_F$, and $\varepsilon_d = \varepsilon_\triangle = \varepsilon^{intra}_\triangle = \varepsilon_X = \frac{\varepsilon}{12}$ for the estimation of degree distributions, triangle counts and $\overline{\Theta}^c_X$.

As we mentioned in Sect. 2, it was shown in [32] that dependencies among tuples can harm the level of privacy offered by differential privacy, since DP implicitly assumes that tuples are independent. To showcase this problem, they showed that in the scenario of differentially private degree sequence computation from an attributed graph with public attributes, node affinity (determined by the similarity of their feature vectors) can be used to improve the adversary's certainty on the existence of edges. For example, users born in the same city, who attended the same university and have approximately the same age are more likely to be connected than the average. Our approach does not suffer from this problem, because we do not treat node attributes as public. Instead, we assign synthetic attribute vectors

to nodes, which are sampled from the differentially private model $\overline{\Theta}_X^c$. In Sect. 4.3, for the sake of efficiency in the synthetic graph generation process, we introduced the assumption that the features of one node are independent from each other. For example, even though intuitively only one of the features "*eye color is blue*", "*eye color is brown*" and "*eye color is green*" can take the value 1 for a given user, we treat them as independent. While this assumption can certainly add inaccuracies in the generated graph, such as a user portrayed as having blue and brown eyes at the same time, it does not create an additional privacy risk, and it does not entail additional assumptions of independence among tuples, other than the ones already implicit in the differentially private methods for computing the model parameters.

6 Experiments

The purpose of our experiments is to empirically validate two claims: (i) our CPGM model outperforms existing models in generating graphs whose community structures are similar to those of the input graphs without sacrificing the ability to preserve global structural properties, and (ii) differentially private instances of CAGM outperform preceding models in terms of community structure preservation, and remain comparable in terms of the preservation of global structural properties.

6.1 Datasets

We use four real-world social networks with node attributes. The first one has been collected from Petster, a website for pet owners [27]. It is an undirected graph, where each node's attributes contain information about the user's pet. We extracted 13 binary attributes from 8 categorical attributes such as favourite food, gender, colour, etc. The second dataset is a subgraph of Facebook available via SNAP [31]. In this dataset, node attributes are already binary and are tagged with serial pseudonyms. For our experiments, we selected the first 50 attributes with the smallest serial numbers. The last two datasets are constructed from a directed graph extracted from Epinions, an online consumer reviews system where every vertex represents a reviewer [37]. In the original dataset, a directed edge from node A to node B exists if user A trusts the reviews of B. For our experiments, we derived an undirected graph from the original dataset by keeping the same vertex set and adding an undirected edge for every pair of mutually trusting users. Additionally, we selected the 50 most frequently rated products as node attributes. If the user rated the product, the value is set to 1. We sampled the subgraph Epinions8K from the undirected version of Epinions by randomly selecting a seed node and progressively taking random neighbouring nodes until totalling 8,000. Table 2 summarises the statistics of the datasets.

Table 2. Datasets used for our experiments.

Dataset	#node	#edges	#△	GCC	#attr.
Petster	1,898	12,534	16,750	0.14	13
Facebook	3,953	84,070	1,526,985	0.54	50
Epinions8K	8,000	67,547	203,257	0.17	50
Epinions	29,515	106,147	235,790	0.13	50

6.2 Evaluation Measures

For every pair (G, G'), where G is a real-life graph and G' is a synthetic graph sampled from a model learned from G, we evaluate the extent to which G' preserves the following properties of G.

Numbers of edges and triangles: Our evaluation measures in this case are the relative errors of the numbers of edges and triangles in G' with respect to G, defined as $\rho_E = \frac{\left||\mathcal{E}_{G'}|-|\mathcal{E}_G|\right|}{|\mathcal{E}_G|}$ and $\rho_\triangle = \frac{|n_\triangle(G')-n_\triangle(G)|}{n_\triangle(G)}$.

Global clustering coefficient: The *global clustering coefficient* (GCC) of a graph measures the proportion of wedges, that is, paths of length 2, that are embedded in triangles. It is defined as $\frac{3n_\triangle}{n_w}$, where n_w is the number of wedges and $n_\triangle$ is the number of triangles. We compare G and G' in terms of the relative error of the GCC of G' with respect to that of G, which we denote ρ_c.

Degree distribution. We compare G and G' in terms of the *Hellinger distance* between their degree distributions. The Hellinger distance has been deemed as the most appropriate distance for comparing probability distributions in previous works on graph synthesising [20, 43]. Given two probability distributions p_1 and p_2 on a discrete domain W, the Hellinger distance between p_1 and p_2 is defined as

$$H(p_1, p_2) = \tfrac{1}{\sqrt{2}}\sqrt{\textstyle\sum_{w\in W}(\sqrt{p_1(w)} - \sqrt{p_2(w)})^2}.$$

The Hellinger distance yields values in $[0, 1]$. The more similar two distributions are, the smaller the Hellinger distance between them. For degree distributions, we compute p_d and p'_d, defined on the domain $W = \{0, 1, \ldots, n-1\}$, where n is the number of vertices in G and G'. For every $i \in W$, $p_d(i)$ (resp. $p'_d(i)$) is the probability that a vertex of G (resp. G') has degree i. The score used for comparing G and G' is $H_d = H(p_d, p'_d)$.

Local clustering coefficients. The *local clustering coefficient* (LCC) of a node v measures the proportion of pairs of mutual neighbours of v that are connected by an edge. In social graphs, $LCC(v)$ is an indicator of the likelihood of v's mutual friends to also be friends. $LCC(v)$ is defined as $\frac{2\sum_{v_i,v_j\in\mathcal{N}(v)} A_{i,j}}{|\mathcal{N}(v)|\cdot(|\mathcal{N}(v)|-1)}$ where $\mathcal{N}(v)$ is the set of v's neighbours. For comparing G and G' in terms of LCC, we compute the distributions $p_{\ell c}$ and $p'_{\ell c}$, which are defined in the domain $W = \{0.01, 0.02, \ldots, 1.0\}$ in such a way that for every $i \in W$, $p_{\ell c}(i)$ (resp. $p'_{\ell c}(i)$) is the probability that a vertex of G (resp. G') has LCC between $i - 0.01$ and i. We compare G and G' in terms of $H(p_{\ell c}, p'_{\ell c})$, and denote this measure by $H_{\ell c}$.

Distribution of attribute-edge correlations. We define, for each community C, the probability distribution p_F^C, where $p_F^C(i)$ is the probability that the similarity between the attribute vectors of two connected nodes in C is i. The original graph G, with community structure $\mathcal{C}$, is compared to a synthetic graph G' in terms of the average of the Hellinger distances of the attribute-edge distributions of all communities of $\mathcal{C}$ in G from those in G', defined as $\rho_a(G, G') = \frac{1}{|\mathcal{C}|}\sum_{C\in\mathcal{C}} H(p_F^C, \widetilde{p}_F^C)$.

Detectability of community partition. We evaluate to what extent state-of-the-art community detection algorithms find similar communities in G and G'. To that end, we use the averaged F_1 score, denoted Avg-F_1, of the community structures $\mathcal{C}$ and $\mathcal{C}'$ determined by the algorithm in G and G', respectively. Given two communities C_1 and C_2, the F_1 score between these two communities, denoted $F_1(C_1, C_2)$ combines two auxiliary measures: *precision* and *recall*. Precision is defined as $prec(C_1, C_2) = \frac{|C_1\cap C_2|}{|C_1|}$, whereas recall is defined as $recall(C_1, C_2) = \frac{|C_1\cap C_2|}{|C_2|}$. Precision and recall are combined as $F_1(C_1, C_2) = \frac{2\cdot prec(C_1,C_2)\cdot recall(C_1,C_2)}{prec(C_1,C_2)+recall(C_1,C_2)}$. If both precision and recall are zero, F_1 is made zero by convention. Following the evaluation strategy introduced in [46, 60, 62], given two sets of communities $\mathcal{C}_1$ and $\mathcal{C}_2$, the average F_1-score is defined as

$$\frac{1}{2|\mathcal{C}_1|}\sum_{C_i^1\in\mathcal{C}_1}\max_{C_j^2\in\mathcal{C}_2} F_1(C_i^1, C_j^2) + \frac{1}{2|\mathcal{C}_2|}\sum_{C_i^2\in\mathcal{C}_2}\max_{C_j^1\in\mathcal{C}_1} F_1(C_i^2, C_j^1).$$

Avg-F_1 values are in $[0, 1]$, and larger Avg-F_1 values indicate more similar community structures.

6.3 Results and Discussion

We first evaluate the ability of our new edge generation model CPGM to synthesise graphs that preserve the community structures of the original graphs along with global structural properties. Then, we assess the overall quality of the differentially private CAGMDP model.

6.3.1 Evaluation of CPGM

We compare CPGM with the two most similar counterparts reported in the literature which are amenable to DP: TriCycle [20] and DCSBM [22]. Fig. 1 shows the extent to which the community structures found by the state-of-the-art community detection method Louvain [4] in the synthetic graphs generated by each model are similar to those detected in the corresponding original graphs. Table 3 compares the behaviours of the edge generation models in terms of global structural properties. In all columns, the values shown are averaged over 100 executions, and smaller values indicate better results.

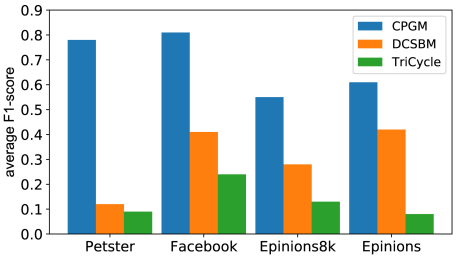


Fig. 1. Similarities of community structures found by Louvain in synthetic graphs to those found in the original graphs.

From the analysis of these results, we extract three major observations. First, as Fig. 1 shows, the community structures of the graphs synthesised using our CPGM model are consistently more similar to those of the original graphs, in comparison to those induced by DCSBM and TriCycle. This supports our claim that CPGM is able to preserve community structure to a larger extent. Note that, in several cases, our model performs almost twice as good as the second best, DCSBM. As expected, TriCycle shows the poorest results, corroborating the intuition that community structure needs to be explicitly included in the generative model if we want synthetic graphs to preserve it. The previous observations support our design choices of preserving (i) the community structure, and (ii) separate intra- and inter-community structural properties. Second, the graphs generated by our CPGM model are consistently the most accurate in terms of the distributions of local clustering coefficients (see right-most column of Table 3), and have close accuracy in terms of global clustering coefficient to the best model (see column labelled ρ_c in Table 3). An analogous observation can be made for the number of triangles (column labelled $\rho_\triangle$). We consider that these observations

Table 3. Comparison of edge generative models in terms of global structural properties.

Dataset	Model	ρ_E	$\rho_\triangle$	ρ_c	H_d	$H_{\ell c}$
Petster	CPGM	**0.00**	**0.18**	**0.05**	**0.16**	**0.19**
	DCSBM	**0.00**	**0.12**	**0.49**	**0.17**	**0.27**
	TriCycle	**0.00**	**0.00**	**0.19**	**0.18**	**0.21**
Facebook	CPGM	**0.00**	**0.03**	**0.32**	**0.15**	**0.32**
	DCSBM	**0.00**	**0.25**	**0.71**	**0.25**	**0.64**
	TriCycle	**0.00**	**0.04**	**0.56**	**0.37**	**0.60**
Epinions8K	CPGM	**0.001**	**0.00**	**0.16**	**0.11**	**0.13**
	DCSBM	**0.001**	**0.63**	**0.81**	**0.12**	**0.40**
	TriCycle	**0.001**	**0.04**	**0.11**	**0.15**	**0.24**
Epinions	CPGM	**0.001**	**0.04**	**0.27**	**0.13**	**0.31**
	DCSBM	**0.002**	**0.60**	**0.83**	**0.14**	**0.26**
	TriCycle	**0.001**	**0.04**	**0.22**	**0.10**	**0.31**

support our design choice of preserving separate intra- and inter-community edge densities and triangle counts. The comparably poorer performance of DCSBM in terms of global and local clustering coefficients also corroborates the need to explicitly model them, as do CPGM and TriCycle. A more detailed graphical description of the behaviour of the three models in terms of the distributions of local clustering coefficients is shown in Fig. 4, in App. F. The figure shows the complementary cumulative distribution functions of local clustering coefficients for the three models, and highlights the special ability of CPGM to capture the behaviour of the distribution for denser-than-normal graphs (the Facebook dataset in this case). Finally, we point out that all three models successfully preserve the properties of the degree distribution. Our CPGM model produces the most consistent results in all the datasets in terms of Hellinger distances.

The results shown in this subsection support our claim that synthetic graphs sampled from CPGM preserve the community structure of the original graph to a considerably larger extent than its closest counterparts, without sacrificing the ability to preserve global structural properties. These results also show that the manner in which CPGM computes intra- and inter-community parameters also helps it outperform competing models in preserving local and global clustering coefficients.

6.3.2 Evaluation of differentially private CAGM

We compare CAGMDP with two other models. The first one is the differentially private AGM model using TriCycle as edge set generator [20]. We refer to this model as AGMDP-Tri. It was shown in [20] that, despite the noise added to guarantee privacy, AGMDP-Tri still preserves to some extent TriCycle's ability to capture global structural properties. As we saw in the previous section, TriCycle performs poorly in preserving the community structure of the original graph, so we consider for our evaluation an additional model, which is a modification of CAGMDP where CPGM is replaced by DCSBM as the edge generation model. We refer to this model as CAGMDP-D.

We compare the behaviour of the three models under four privacy budgets: 2.0, 3.0, 4.0 and 5.0. We chose these values to ensure that the part of the privacy budget devoted to ModDivisive (1.0, 1.5, 2.0 and 2.5, respectively) is within the same range as the budgets used in the experiments reported in [46]. The authors of ModDivisive state that epsilon values for their method should be in these ranges in order to enable the accurate privacy-preserving generation of a search tree that guarantees to yield communities of reasonable size. The use of smaller values usually leads to obtaining a single community containing all nodes, which is equivalent to obtaining no community partition at all. Each comparison between instances of the three models uses the same value for the privacy budget. Since CAGMDP-D and AGMDP-Tri each have fewer parameters than CAGMDP, we re-allocate in each case the remaining privacy budget to other computations. In estimating CAGMDP-D, we allocate to community partition the same budget as for CAGMDP, i.e. $\frac{\varepsilon}{2}$. CAGMDP-D requires to compute the numbers of edges between every pair of communities. We assign to this computation the budget used in CAGMDP for counting the number of intra-community triangles, i.e. $\frac{\varepsilon}{12}$. Finally, CAGMDP-D is given for degree sequence computation the budget $\varepsilon'_d = \varepsilon_d + \varepsilon_\triangle = \frac{\varepsilon}{6}$, as it does not require to count the global number of triangles. Since AGMDP-Tri computes every parameter computed by CAGMDP, except for the community partition (which takes half of the budget of CAGMDP) we double the budget assigned to every other computation of AGMDP-Tri. These re-allocations give comparative advantages to both competing models CAGMDP-D and AGMDP-Tri in their comparison with our model, as they will be able to more accurately compute some of the parameters they have in common with our model. We allow this advantage because requiring a smaller number of computations is a positive feature of differentially private methods, so it should not be punished in the comparison. In CAGMDP and CAGMDP-D, ModDivisive is run with $w_s = 0.98$. Finally, in estimating the attribute-edge correlations distribution (discussed in Sect. 5.3), we set the maximum degree parameter p to 100.

Fig. 2 displays the behaviours of the three models in terms of community structure preservation, whereas Tables 4, 5, 6 and 7 summarise their behaviours in terms

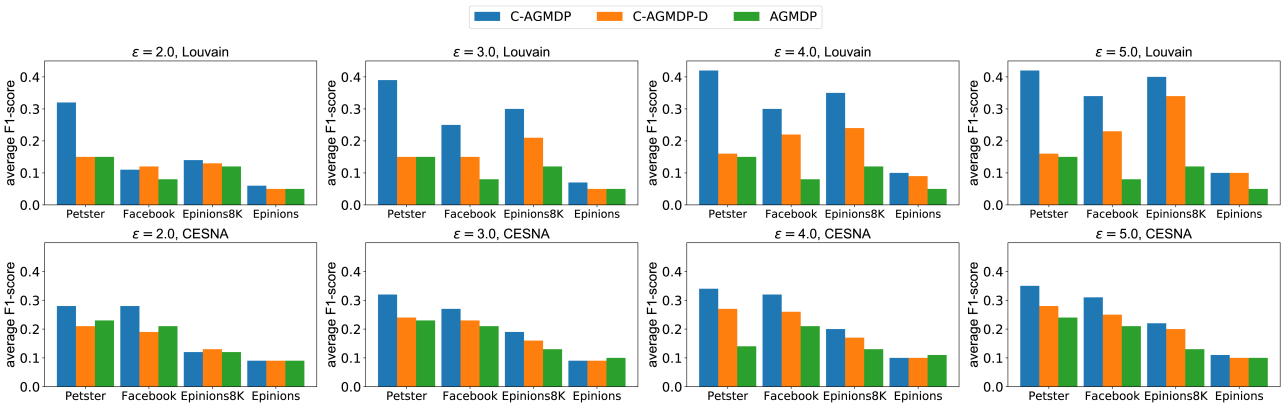


Fig. 2. Comparison of differentially private models in terms of community structure preservation.

of global structural properties and attribute-edge correlations on the selected datasets. In what follows, we analyse these results from three different perspectives.

Community structure preservation. In Fig. 2, the four upper charts display the extent to which the community structures found by Louvain in the synthetic graphs generated by each differentially private model are similar to those detected in the corresponding original graphs. Two important features of the Louvain algorithm are shared by ModDivisive, the method used for obtaining community partitions in CAGMDP and CAGMDP-D. Both generate a community partition, and both operate by maximising modularity. In order to assess whether the community structures induced by our models in the synthetic graphs are also detectable by algorithms based on different criteria, we additionally obtained analogous results using the algorithm CESNA [62]. These results are shown in the lower four charts of Fig. 2. Unlike Louvain, CESNA takes node attributes into consideration for computing communities. However, CESNA tends to obtain substantially overlapping communities, whereas both CAGMDP and CAGMDP-D assume a partition. CESNA requires as a parameter the number of communities, which we set to 10.

From the analysis of these results, the most relevant observation is that, considering the communities detected by both Louvain and CESNA, the graphs sampled from our CAGMDP model consistently rank as the ones whose community structure is most similar to that of the corresponding original graphs. In the particular cases of Petster, Facebook and Epinions8K with the Louvain algorithm, the similarity values displayed by our CAGMDP model are in most cases more than twice better than their counterparts for AGMDP-Tri, and considerably more than those for CAGMDP-D. However, this advantage decreases as the number of vertices grows. All three models fail to effectively preserve the community structures on Epinions, which has more than $20,000$ nodes. This is because the amount of noise added to enforce DP grows proportionally to the number of nodes.

Distributions of attribute-edge correlations. For each original graph, we compute the distributions of attribute-edge correlations in all communities detected by CESNA. Then, we compute the equivalent distributions on each synthetic graph and compare them to that of the original graph in terms of ρ_a (see right-most columns of Tables 4, 5, 6 and 7). From the analysis of these results, we can see that the synthetic graphs sampled from CAGMDP and CAGMDP-D, the two models that consider community structures, consistently outperform those sampled from AGMDP-Tri in terms of ρ_a. Another important observation is that the qualities, in terms of ρ_a, of synthetic graphs sampled from our CAGMDP model and those sampled from its variant CAGMDP-D are quite similar. This observation suggests that CAGMDP can in some cases be seen as a *meta-model*, where several edge generation models, e.g. CPGM and DCSBM, can be used.

Global structural properties. From the analysis of Tables 4, 5, 6 and 7, we can see that, as expected, our CAGMDP model suffers a larger degradation than CAGMDP-D and AGMDP-Tri in terms of the measures that depend on parameters for which the latter models were allocated larger privacy budgets, most notably the numbers of edges. The reason for these relatively larger errors is that our edge generative model CPGM requires to perturb two degree-related parameters (intra- and inter-community degrees) while other models just have one. This also affects the performance of CAGMDP in preserving global clustering coefficients.

It is worth noting, however, that in the cases where a differentially private parameter underwent a post-processing, most notably regarding the number of tri-

Table 4. Comparison of DP models on Petster.

ε	Model	ρ_E	$\rho_\triangle$	ρ_c	H_d	$H_{\ell c}$	ρ_a
2.0	CAGMDP	0.56	0.09	0.43	0.42	0.39	0.14
	CAGMDP-D	0.22	0.55	0.69	0.30	0.44	0.23
	AGMDP-Tri	0.25	0.09	0.25	0.23	0.30	0.17
3.0	CAGMDP	0.31	0.09	0.23	0.33	0.29	0.16
	CAGMDP-D	0.11	0.30	0.55	0.24	0.34	0.16
	AGMDP-Tri	0.13	0.09	0.21	0.19	0.26	0.17
4.0	CAGMDP	0.19	0.08	0.20	0.29	0.26	0.13
	CAGMDP-D	0.06	0.29	0.54	0.22	0.32	0.14
	AGMDP-Tri	0.09	0.08	0.19	0.19	0.25	0.17
5.0	CAGMDP	0.13	0.08	0.08	0.24	0.23	0.09
	CAGMDP-D	0.04	0.29	0.54	0.20	0.31	0.12
	AGMDP-Tri	0.06	0.10	0.17	0.18	0.24	0.16

Table 5. Comparison of DP models on Facebook.

ε	Model	ρ_E	$\rho_\triangle$	ρ_c	H_d	$H_{\ell c}$	ρ_a
2.0	CAGMDP	0.15	0.04	0.65	0.32	0.59	0.13
	CAGMDP-D	0.07	0.89	0.90	0.19	0.89	0.08
	AGMDP-Tri	0.09	0.08	0.58	0.31	0.58	0.19
3.0	CAGMDP	0.12	0.04	0.59	0.18	0.47	0.09
	CAGMDP-D	0.05	0.63	0.79	0.17	0.76	0.06
	AGMDP-Tri	0.07	0.09	0.58	0.32	0.58	0.19
4.0	CAGMDP	0.09	0.04	0.53	0.17	0.44	0.07
	CAGMDP-D	0.05	0.39	0.73	0.16	0.69	0.05
	AGMDP-Tri	0.06	0.09	0.59	0.32	0.58	0.08
5.0	CAGMDP	0.07	0.04	0.52	0.17	0.43	0.06
	CAGMDP-D	0.04	0.37	0.71	0.16	0.68	0.04
	AGMDP-Tri	0.05	0.09	0.59	0.33	0.58	0.19

Table 6. Comparison of DP models on Epinions8K.

ε	Model	ρ_E	$\rho_\triangle$	ρ_c	H_d	$H_{\ell c}$	ρ_a
2.0	CAGMDP	0.15	0.15	0.44	0.34	0.34	0.15
	CAGMDP-D	0.07	0.82	0.86	0.32	0.49	0.14
	AGMDP-Tri	0.09	0.06	0.10	0.20	0.24	0.19
3.0	CAGMDP	0.12	0.15	0.61	0.25	0.30	0.14
	CAGMDP-D	0.05	0.76	0.83	0.26	0.44	0.14
	AGMDP-Tri	0.07	0.08	0.09	0.17	0.22	0.18
4.0	CAGMDP	0.09	0.11	0.54	0.20	0.23	0.14
	CAGMDP-D	0.05	0.71	0.81	0.22	0.42	0.13
	AGMDP-Tri	0.06	0.09	0.11	0.17	0.21	0.18
5.0	CAGMDP	0.07	0.07	0.48	0.17	0.20	0.14
	CAGMDP-D	0.04	0.57	0.76	0.18	0.35	0.13
	AGMDP-Tri	0.05	0.09	0.13	0.15	0.21	0.18

Table 7. Comparison of DP models on Epinions.

ε	Model	ρ_E	$\rho_\triangle$	ρ_c	H_d	$H_{\ell c}$	ρ_a
2.0	CAGMDP	1.29	0.32	0.64	0.55	0.40	0.09
	CAGMDP-D	0.55	0.95	0.96	0.50	0.32	0.08
	AGMDP-Tri	0.17	0.18	0.19	0.33	0.37	0.15
3.0	CAGMDP	0.79	0.38	0.66	0.44	0.25	0.09
	CAGMDP-D	0.30	0.84	0.84	0.35	0.28	0.08
	AGMDP-Tri	0.13	0.25	0.08	0.28	0.32	0.14
4.0	CAGMDP	0.53	0.29	0.56	0.34	0.22	0.08
	CAGMDP-D	0.18	0.89	0.92	0.33	0.27	0.08
	AGMDP-Tri	0.11	0.28	0.02	0.24	0.30	0.14
5.0	CAGMDP	0.38	0.23	0.53	0.27	0.21	0.08
	CAGMDP-D	0.14	0.85	0.90	0.30	0.27	0.08
	AGMDP-Tri	0.08	0.21	0.08	0.18	0.28	0.14

angles, our model obtained error rates of the same scale as AGMDP-Tri. For example, the error rate dropped to 0.04 for the Facebook dataset. Also in the Facebook graph, despite the larger error rate in the number of edges, in some cases our model showed roughly the same or even better performance in preserving the degree sequence and clustering coefficients than the AGMDP-Tri model. This is also reflected in preserving the density of node neighbourhoods. CAGMDP has the same power as AGMDP-Tri in preserving local clustering coefficients. Finally, since the amount of added noise grows with the number of vertices, synthetic graphs generated by all three models for the Epinions dataset suffer from a more significant degradation in terms of structural properties.

7 Conclusions

We have presented, to the best of our knowledge, the first community-preserving differentially private method for publishing synthetic attributed graphs. To devise this method, we developed CAGM, a new community-preserving generative attributed graph model. We have equipped CAGM with efficient parameter estimation and sampling methods, and have devised differentially private variants of the former. A comprehensive set of experiments on real-world datasets support the claim that our method is able to generate useful synthetic graphs satisfying a strong formal privacy guarantee. Our main direction for future work is to improve CAGM by increasing the repertoire of community-related statistics captured by the model, and by equipping it with a new differentially private community partition method that integrates node attributes via a low-sensitivity objective function and/or differentially private maximum-likelihood estimation methods.

8 Acknowledgements

This work was funded by Luxembourg's Fonds National de la Recherche, via grant C17/IS/11685812 (PrivDA).

References

[1] Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation). *OJ*, L 119:1–88, 4.5.2016.

[2] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Review of Modern Physics*, 74:47–97, 2002.

[3] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. Differentially private data analysis of social networks via restricted sensitivity. In *Proc. 4th Innovations in Theoretical Computer Science (ITCS)*, pages 87–96. ACM Press, 2013.

[4] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[5] Jordi Casas-Roma, Jordi Herrera-Joancomartí, and Vicenç Torra. An algorithm for k-degree anonymity on large networks. In *Procs. of the 2013 IEEE/ACM Int'l Conf. on Advances in Social Networks Analysis and Mining*, pages 671–675, 2013.

[6] Jordi Casas-Roma, Jordi Herrera-Joancomartí, and Vicenç Torra. Anonymizing graphs: measuring quality for clustering. *Knowledge and Information Systems*, 44(3):507–528, 2015.

[7] Jordi Casas-Roma, Jordi Herrera-Joancomartí, and Vicenç Torra. k-degree anonymity and edge selection: improving data utility in large networks. *Knowledge and Information Systems*, 50(2):447–474, 2017.

[8] Lauren E. Charles-Smith, Tera L. Reynolds, Mark A. Cameron, Mike Conway, Eric H. Y. Lau, Jennifer M. Olsen, Julie A. Pavlin, Mika Shigematsu, Laura C. Streichert, Katie J. Suda, and Courtney D. Corley. Using social media for actionable disease surveillance and outbreak management: A systematic literature review. *PLOS ONE*, 10(10):1–20, 10 2015.

[9] James Cheng, Ada Wai-chee Fu, and Jia Liu. K-isomorphism: privacy preserving network publication against structural attacks. In *Procs. of the 2010 ACM SIGMOD Int'l Conf. on Management of Data*, pages 459–470, 2010.

[10] Sean Chester, Bruce M Kapron, Ganesh Ramesh, Gautam Srivastava, Alex Thomo, and S Venkatesh. Why waldo befriended the dummy? k-anonymization of social networks with pseudo-nodes. *Social Network Analysis and Mining*, 3(3):381–399, 2013.

[11] Fan Chung and Linyuan Lu. The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences*, 99(25):15879–15882, 2002.

[12] Aaron Clauset, Cristopher Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453:98–101, 2008.

[13] Cynthia Dwork. Differential privacy. In *Proc. 33rd International Colloquium on Automata, Languages and Programming (ICALP)*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006.

[14] Luoyi Fu, Xinzhe Fu, Zhongzhao Hu, Zhiying Xu, and Xinbing Wang. De-anonymization of social networks with communities: When quantifications meet algorithms. *arXiv preprint arXiv:1703.09028*, 2017.

[15] Michael Hay, Chao Li, Gerome Miklau, and David D. Jensen. Accurate estimation of the degree distribution of private networks. In *Proc. 19th IEEE International Conference on Data Mining (ICDM)*, pages 169–178. IEEE Computer Society, 2009.

[16] Michael Hay, Gerome Miklau, David D. Jensen, Donald F. Towsley, and Philipp Weis. Resisting structural re-identification in anonymized social networks. *PVLDB*, 1(1):102–114, 2008.

[17] Joseph J. Pfeiffer III, Timothy La Fond, Sebastián Moreno, and Jennifer Neville. Fast generation of large scale social networks while incorporating transitive closures. In *Proc. 4th International Conference on Privacy, Security, Risk and Trust, (PASSAT)*, pages 154–165. IEEE Computer Society, 2012.

[18] Joseph J. Pfeiffer III, Sebastián Moreno, Timothy La Fond, Jennifer Neville, and Brian Gallagher. Attributed graph models: modeling network structure with correlated attributes. In *Proc. 23rd International World Wide Web Conference (WWW)*, pages 831–842. ACM Press, 2014.

[19] Shouling Ji, Weiqing Li, Prateek Mittal, Xin Hu, and Raheem Beyah. Secgraph: A uniform and open-source evaluation system for graph data anonymization and de-anonymization. In *24th {USENIX} Security Symposium ({USENIX} Security 15)*, pages 303–318, 2015.

[20] Zach Jorgensen, Ting Yu, and Graham Cormode. Publishing attributed social graphs with formal privacy guarantees. In *Proc. 2016 International Conference on Management of Data (SIGMOD)*, pages 107–122. ACM Press, 2016.

[21] Kundan Kandhway and Joy Kuri. Using node centrality and optimal control to maximize information diffusion in social networks. *IEEE Trans. Systems, Man, and Cybernetics: Systems*, 47(7):1099–1110, 2017.

[22] Brian Karrer and Mark EJNewman. Stochastic blockmodels and community structure in networks. *Physical review. E*, 83(1):016107, 2011.

[23] Vishesh Karwa, Sofya Raskhodnikova, Adam D. Smith, and Grigory Yaroslavtsev. Private analysis of graph structure. *ACM Transactions on Database Systems*, 39(3):22:1–22:33, 2014.

[24] Vishesh Karwa and Aleksandra B. Slavkovic. Differentially private graphical degree sequences and synthetic graphs. In *Proc. 2012 International Conference on Privacy in Statistical Databases (PSD)*, volume 7556 of *Lecture Notes in Computer Science*, pages 273–285. Springer, 2012.

[25] Daniel Kifer and Bing-Rong Lin. Towards an axiomatization of statistical privacy and utility. In *Proc. 29th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, pages 147–158. ACM Press, 2010.

[26] Tamara G. Kolda, Ali Pinar, Todd D. Plantenga, and C. Seshadhri. A scalable generative graph model with community structure. *SIAM J. Scientific Computing*, 36(5), 2014.

[27] Jérôme Kunegis. KONECT: the koblenz network collection. In *Proc. 22nd International World Wide Web Conference (WWW)*, pages 1343–1350. ACM Press, 2013.

[28] Christine Largeron, Pierre-Nicolas Mougel, Oualid Benyahia, and Osmar R Zaïane. Dancer: dynamic attributed networks with community structure generation. *Knowledge and Information Systems*, 53(1):109–151, 2017.

[29] Christine Largeron, Pierre-Nicolas Mougel, Reihaneh Rabbany, and Osmar R Zaïane. Generating attributed networks with communities. *PloS one*, 10(4), 2015.

[30] Jure Leskovec and Christos Faloutsos. Scalable modeling of real graphs using kronecker multiplication. In *Proc. 24th International Conference on Machine Learning (ICML)*, pages 497–504. ACM Press, 2007.

[31] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, 2014.

[32] Changchang Liu, Supriyo Chakraborty, and Prateek Mittal. Dependence makes you vulnberable: Differential privacy under dependent tuples. In *Procs. of NDSS 2016*, volume 16, pages 21–24, 2016.

[33] Kun Liu and Evimaria Terzi. Towards identity anonymization on graphs. In *Proc. 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 93–106. ACM Press, 2008.

[34] Xuesong Lu, Yi Song, and Stéphane Bressan. Fast identity anonymization on graphs. In *Procs. of the Int'l Conf. on Database and Expert Systems Applications*, pages 281–295, 2012.

[35] Tinghuai Ma, Yuliang Zhang, Jie Cao, Jian Shen, Meili Tang, Yuan Tian, Abdullah Al-Dhelaan, and Mznah Al-Rodhaan. Kdvem: a k-degree anonymity with vertex and edge modification algorithm. *Computing*, 97(12):1165–1184, 2015.

[36] Nelly Marquetoux, Mark A. Stevenson, Peter Wilson, Anne Ridler, and Cord Heuer. Using social network analysis to inform disease control interventions. *Preventive Veterinary Medicine*, 126:94–104, 2016.

[37] Paolo Massa and Paolo Avesani. Trust-aware recommender systems. In *Proc. 2007 ACM Conference on Recommender Systems (RecSys)*, pages 17–24. ACM Press, 2007.

[38] Sjouke Mauw, Yunior Ramírez-Cruz, and Rolando Trujillo-Rasua. Anonymising social graphs in the presence of active attackers. *Transactions on Data Privacy*, 11(2):169–198, 2018.

[39] Sjouke Mauw, Yunior Ramírez-Cruz, and Rolando Trujillo-Rasua. Conditional adjacency anonymity in social graphs under active attacks. *Knowledge and Information Systems*, 61(1):485–511, 2018.

[40] Frank McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. *Communications of the ACM*, 53(9):89–97, 2010.

[41] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Proc. 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 94–103. IEEE Computer Society, 2007.

[42] Darakhshan J. Mir and Rebecca N. Wright. A differentially private graph estimator. In *Proc. 2009 ICDM International Workshop on Privacy Aspects of Data Mining (ICDM)*, pages 122–129. IEEE Computer Society, 2009.

[43] Prateek Mittal, Charalampos Papamanthou, and Dawn Xiaodong Song. Preserving link privacy in social network based systems. In *Procs. of NDSS 2013*. The Internet Society, 2013.

[44] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Reviewe E*, 69(2):026113, 2004.

[45] Mark E. J. Newman. Community detection in networks: Modularity optimization and maximum likelihood are equivalent. *CoRR*, abs/1606.02319, 2016.

[46] Hiep H. Nguyen, Abdessamad Imine, and Michaël Rusinowitch. Detecting communities under differential privacy. In *Proc. 2016 ACM Workshop on Privacy in the Electronic Society (WPES)*, pages 83–93. ACM Press, 2016.

[47] Shirin Nilizadeh, Apu Kapadia, and Yong-Yeol Ahn. Community-enhanced de-anonymization of online social networks. In *Proceedings of the 2014 acm sigsac conference on computer and communications security*, pages 537–548, 2014.

[48] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proc. 39th Annual ACM Symposium on Theory of Computing (STOC)*, pages 75–84. ACM Press, 2007.

[49] Liudmila Ostroumova Prokhorenkova and Alexey Tikhonov. Community detection through likelihood optimization: In search of a sound model. In *Proc. 30th World Wide Web Conference (WWW)*, pages 1498–1508. ACM Press, 2019.

[50] François Rousseau, Jordi Casas-Roma, and Michalis Vazirgiannis. Community-preserving anonymization of graphs. *Knowledge and Information Systems*, 54(2):315–343, 2017.

[51] Alessandra Sala, Xiaohan Zhao, Christo Wilson, Haitao Zheng, and Ben Y. Zhao. Sharing graphs using differentially private graph models. In *Proc. 11th ACM SIGCOMM Internet Measurement Conference (IMC)*, pages 81–98. ACM Press, 2011.

[52] Julián Salas and Vicenç Torra. Graphic sequences, distances and k-degree anonymity. *Discrete Applied Mathematics*, 188:25–31, 2015.

[53] Yazhe Wang, Long Xie, Baihua Zheng, and Ken CK Lee. High utility k-anonymization for social network publishing. *Knowledge and Information Systems*, 41(3):697–725, 2014.

[54] Yue Wang and Xintao Wu. Preserving differential privacy in degree-correlation based graph generation. *Transaction on Data Privacy*, 6(2):127–145, 2013.

[55] Yue Wang, Xintao Wu, Jun Zhu, and Yang Xiang. On learning cluster coefficient of private networks. *Social Network Analysis and Mining*, 3(4):925–938, 2013.

[56] Paul W.Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

[57] Gilbert Wondracek, Thorsten Holz, Engin Kirda, and Christopher Kruegel. A practical attack to de-anonymize social network users. In *2010 IEEE Symposium on Security and Privacy*, pages 223–238. IEEE, 2010.

[58] Wentao Wu, Yanghua Xiao, Wei Wang, Zhenying He, and Zhihui Wang. K-symmetry model for identity anonymization in social networks. In *Procs. of the 13th Int'l Conf. on Extending Database Technology*, pages 111–122, 2010.

[59] Qian Xiao, Rui Chen, and Kian-Lee Tan. Differentially private network data release via structural inference. In *Proc. 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 911–920. ACM Press, 2014.

[60] Jaewon Yang and Jure Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proc. 6th ACM International Conference on Web Search and Data Mining (WSDM)*, pages 587–596. ACM Press, 2013.

[61] Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213, 2015.

[62] Jaewon Yang, Julian J. McAuley, and Jure Leskovec. Community detection in networks with node attributes. In *Proc. 13th IEEE International Conference on Data Mining (ICDM)*, pages 1151–1156. IEEE Computer Society, 2013.

[63] Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Private release of graph statistics using ladder functions. In *Proc. 36th ACM International Conference on Management of Data (SIGMOD)*, pages 731–745. ACM Press, 2015.

[64] Yang Zhang, Mathias Humbert, Bartlomiej Surma, Praveen Manoharan, Jilles Vreeken, and Michael Backes. Towards plausible graph anonymization. In *Procs. of NDSS 2020*, 2020.

[65] Bin Zhou and Jian Pei. The *k*-anonymity and *l*-diversity approaches for privacy preservation in social networks against neighborhood attacks. *Knowledge Information Systems*, 28(1):47–77, 2011.

[66] Dmitry Zinoviev. *Information Diffusion in Social Networks*, pages 146–163. 11 2011.

[67] Lei Zou, Lei Chen, and M. Tamer Özsu. K-automorphism: A general framework for privacy preserving network publication. *PVLDB*, 2(1):946–957, 2009.

A Proof of Proposition 1

Proposition 1. *Every graph G of order n satisfies $LS_{Q_a(\mathcal{C})}(G) \leq \frac{60}{n}$.*

Proof. Let $G \sim_{at} G'$ be two neighbouring attributed graphs and let G_a and G'_a be the auxiliary graphs obtained from G and G', respectively. If the difference between G and G' consists only in one edge, then $G_a = G'_a$, so in what follows we will consider that G and G' differ in one attribute vector. Let v be the (sole) vertex such that $\tau_G(v) \neq \tau_{G'}(v)$. In the worst case, we have that, for every $w \in \mathcal{V} \setminus \{v\}$, $(v,w) \in G_a$ and $(v,w) \notin G'_a$ (or *vice versa*). It was shown in [46] that the modularities of two graphs differing in one edge differ in up to $\frac{3}{m}$, where m is the minimum number of edges. Then, in the worst case we have $LS_{Q_a}(G) \leq \frac{3(n-1)}{m_a}$, where n is the order of G and G', and m_a is the minimum number of edges in auxiliary graphs. As we discussed in Sect. 5.1, $m_a \geq \frac{n(n-1)}{20}$, so $LS_{Q_a}(G) \leq \frac{60}{n}$. The proof is thus completed. □

B Proof of Proposition 2

Proposition 2. *The global sensitivity of n_C is k.*

Proof. Let $G \sim_{at} G'$ be two neighbouring attributed graphs, let $C \subseteq \mathcal{V}$ be a community and let $n_C(G)$ and $n_C(G')$ be the instances of $n_C = \left(n_C^1, n_C^2, \ldots, n_C^k\right)$ in G and G', respectively. If the difference between G and G' consists only in one edge, then $n_C(G) = n_C(G')$, so in what follows we will consider that G and G' differ in one attribute vector. Let v be the (sole) vertex such that $\tau_G(v) \neq \tau_{G'}(v)$. If $v \notin C$, then $n_C(G) = n_C(G')$. On the contrary, if $v \in C$, for every component $\ell \in \{1, \ldots, k\}$ such that $\tau_{\ell,G}(v) \neq \tau_{\ell,G'}(v)$, we have that $\left|n_C^\ell(G) - n_C^\ell(G')\right| = 1$. In consequence, we have $\Delta_{n_C} = \max_{G \sim_{at} G'} \left\|n_C(G) - n_C(G')\right\|_1 = k$. □

C Proof of Proposition 3

Proposition 3. *The global sensitivity of the number of intra-community triangles of a graph G with a community partition $\mathcal{C}$ is $\Delta_{n_\triangle^{intra}} = \max_{C \in \mathcal{C}}\{|C| - 2\}$.*

Proof. Let $G \sim_{at} G'$ be two neighbouring attributed graphs and let $\mathcal{C}$ be a community partition of $\mathcal{V}$. Let $n_\triangle^{intra}(G)$ and $n_\triangle^{intra}(G')$ be the numbers of intra-community triangles of G and G', respectively. If the difference between G and G' consists only in one attribute vector, then $n_\triangle^{intra}(G) = n_\triangle^{intra}(G')$, so in what follows we will consider that G and G' differ in one edge. We will assume, without loss of generality, that $\mathcal{E}' \setminus \mathcal{E} = (v, v')$. Two cases are possible for $\psi_{\mathcal{C}}(v)$ and $\psi_{\mathcal{C}}(v')$:

(i) $\psi_{\mathcal{C}}(v) \neq \psi_{\mathcal{C}}(v')$. In this case, since (v, v') is an inter-community edge, $n_\triangle^{intra}(G) = n_\triangle^{intra}(G')$.

(ii) $\psi_{\mathcal{C}}(v) = \psi_{\mathcal{C}}(v') = C$. In this case, we have that $n_\triangle^{intra}(G') - n_\triangle^{intra}(G) = |C \cap \mathcal{N}_G(v) \cap \mathcal{N}_G(v')|$, that is the number of common neighbours of v and v' in the same community.

It is simple to see that every pair of vertices v and v' such that $\psi_{\mathcal{C}}(v) = \psi_{\mathcal{C}}(v') = C$ satisfy $|C \cap \mathcal{N}_G(v) \cap \mathcal{N}_G(v')| \leq |C| - 2$. Hence, $\Delta_{n_\triangle^{intra}} = \max_{C \in \mathcal{C}}\{|C| - 2\}$. □

D Proof of Proposition 4

Proposition 4. *For every graph G, every community partition $\mathcal{C}$ of G, and every positive integer $t \geq 1$,*

$$LS_{n_\triangle^{intra}}((G,\mathcal{C}),t) = \max_{\{i,j \mid \psi_\mathcal{C}(v_i)=\psi_\mathcal{C}(v_j)\}} \left\{ \min\left\{ a_{ij} + \left\lfloor \frac{t+\min\{t,b_{ij}\}}{2} \right\rfloor, |\psi_\mathcal{C}(v_i)| - 2 \right\} \right\},$$

where $a_{ij} = |\{v_\ell \in \psi_\mathcal{C}(v_i) \mid A_{i,\ell} = 1 \wedge A_{j,\ell} = 1\}|$ and $b_{ij} = |\{v_\ell \in \psi_\mathcal{C}(v_i) \mid A_{i,\ell} \oplus A_{j,\ell} = 1\}|$.

Proof. Consider a graph G with a community partition $\mathcal{C}$, and a positive integer $t \geq 1$. As discussed in [48, 63],

$$LS_{n_\triangle^{intra}}((G,\mathcal{C}),t) = \max_{1 \leq i < j \leq |\mathcal{V}|} LS_{ij}^{n_\triangle^{intra}}((G,\mathcal{C}),t).$$

For every i and j such that $\psi_\mathcal{C}(v_i) \neq \psi_\mathcal{C}(v_j)$, we have that no intra-community triangle is created (resp. destroyed) by the addition (resp. removal) of (v_i, v_j), so $LS_{ij}^{n_\triangle^{intra}}((G,\mathcal{C}),t) = 0$. Thus,

$$LS_{n_\triangle^{intra}}((G,\mathcal{C}),t) = \max_{\{i,j \mid \psi_\mathcal{C}(v_i)=\psi_\mathcal{C}(v_j)\}} LS_{ij}^{n_\triangle^{intra}}((G,\mathcal{C}),t).$$

We now focus on determining $LS_{ij}^{n_\triangle^{intra}}((G,\mathcal{C}),t)$ for every i and j such that $\psi_\mathcal{C}(v_j) = \psi_\mathcal{C}(v_j) = C$. Consider a pair of such values i and j, and let $S_1 = \{v_\ell \in C \mid A_{i,\ell} = 1 \wedge A_{j,\ell} = 1\}$ and $S_2 = \{v_\ell \in C \mid A_{i,\ell} \oplus A_{j,\ell} = 1\}$[1].

Let $\mathcal{G}_t$ be the class of all graphs that can be obtained by modifying G as follows:

1. Add $\min\{b_{ij}, t\}$ arbitrary edges of the form (x, y), where $x \in \{v_i, v_j\}$ and $y \in S_2$.
2. If $t > b_{ij}$, take an arbitrary subset S_3 of vertices of $C \setminus (S_1 \cup S_2 \cup \{v_i, v_j\})$, with cardinality $\min\left\{ \left\lfloor \frac{t-b_{ij}}{2} \right\rfloor, |C \setminus (S_1 \cup S_2 \cup \{v_i, v_j\})| \right\}$. For every $x \in S_3$, add the edges (v_i, x) and (v_j, x).

From the definition of $\mathcal{G}_t$, it follows that every $G' \in \mathcal{G}_t$ satisfies $\phi(G, G') \leq t$ and the graph $G'' \sim_{ij} G'$ satisfies

$$|n_\triangle^{intra}(G') - n_\triangle^{intra}(G'')| = \min\left\{ a_{ij} + \left\lfloor \frac{t+\min\{t,b_{ij}\}}{2} \right\rfloor, |C| - 2 \right\}.$$

Now, consider an arbitrary graph G', obtained by modifying G, such that $\phi(G, G') \leq t$ and $G' \notin \mathcal{G}_t$. Also consider the graph $G'' \sim_{ij} G'$. According to the definition of $\mathcal{G}_t$, the following situations are possible:

(i) G' is the result of adding to G a proper subset of the set of edges added by steps 1 and 2 of the procedure described above for obtaining an element of $\mathcal{G}_t$. In this case, only a proper subset of the triangles created (resp. destroyed) by the addition (resp. removal) of (x, y) is added (resp. removed). Thus,

$$|n_\triangle^{intra}(G') - n_\triangle^{intra}(G'')| < \min\left\{ a_{ij} + \left\lfloor \frac{t+\min\{t,b_{ij}\}}{2} \right\rfloor, |C| - 2 \right\}.$$

(ii) G' is the result of applying $t - t'$ additional modifications ($t' < t$) on an element H of $\mathcal{G}_t$. Note that, by the definition of edge-edit distance, the additional modifications do not consist in reverting any edge addition made in steps 1 and 2 of the procedure described above. In this case, none of the additional modifications can result in the addition of a par of edges of the form (v_i, x) and (v_j, x), with $x \in C$, so

$$|n_\triangle^{intra}(G') - n_\triangle^{intra}(G'')| = |n_\triangle^{intra}(H) - n_\triangle^{intra}(H')| = \min\left\{ a_{ij} + \left\lfloor \frac{t+\min\{t,b_{ij}\}}{2} \right\rfloor, |C| - 2 \right\},$$

where $H' \sim_{ij} H$.

(iii) In every other case, the transformation that allows to obtain G' from G can be divided into a set of edge additions as the ones described in (i) and a set of additional modifications as the ones described in (ii). Applying an analogous reasoning, we have that

$$|n_\triangle^{intra}(G') - n_\triangle^{intra}(G'')| < \min\left\{ a_{ij} + \left\lfloor \frac{t+\min\{t,b_{ij}\}}{2} \right\rfloor, |C| - 2 \right\}.$$

Summing up the set of cases analysed above, we have that, for every i and j such that $\psi_\mathcal{C}(v_i) = \psi_\mathcal{C}(v_j)$,

$$LS_{ij}^{n_\triangle^{intra}}((G,\mathcal{C}),t) = \max_{\{G', G'' \mid \phi(G,G') \leq t, G' \sim_{ij} G''\}} \{|n_\triangle^{intra}(G') - n_\triangle^{intra}(G'')|\} = \min\left\{ a_{ij} + \left\lfloor \frac{t+\min\{t,b_{ij}\}}{2} \right\rfloor, |\psi_\mathcal{C}(v_i)| - 2 \right\}$$

and, in consequence,

$$LS_{n_\triangle^{intra}}((G,\mathcal{C}),t) = \max_{\{i,j \mid \psi_\mathcal{C}(v_i)=\psi_\mathcal{C}(v_j)\}} LS_{ij}^{n_\triangle^{intra}}((G,\mathcal{C}),t) = \max_{\{i,j \mid \psi_\mathcal{C}(v_i)=\psi_\mathcal{C}(v_j)\}} \left\{ \min\left\{ a_{ij} + \left\lfloor \frac{t+\min\{t,b_{ij}\}}{2} \right\rfloor, |\psi_\mathcal{C}(v_i)| - 2 \right\} \right\}.$$

The proof is thus completed. □

[1] Note that S_1 and S_2 are the sets whose cardinalities define a_{ij} and b_{ij}, respectively

E Resistance against community-enhanced re-identification attacks

We implemented the strongest community-enhanced re-identification attack reported in the literature [47], following the specification given in the paper. Fig. 3 shows the success rate of this attack on pseudonymised versions of synthetic graphs generated by our model. In the figure, dashed lines represent the success rates of the attack on the original graphs, whereas solid lines represent the success rates on the pseudonymised synthetic graphs. The figure clearly shows that, in the best cases, the attack identifies around 5% of users in the pseudonymised synthetic graphs generated by our approach. These values are considerably low in comparison with the success rates on the original graphs, which range from 65% to 75%. The success rates obtained by the attack on the original graphs in these experiments are consistent with the ones reported in [47]. Regarding the attack's ineffectiveness on synthetic graphs, it stems from its underlying assumption that, after anonymisation, the number of changes in the neighbourhood of each individual vertex is relatively small. This assumption does not hold in the scenario of graph synthesis. In this scenario, the methods focus on generating graphs that are similar to the original one in terms of global properties. However, the synthetic edge sets generally differ from the original ones to a larger extent than assumed by the attack. This behaviour effectively thwarts the attack, and is rather independent of epsilon values, as illustrated in Fig. 3. In fact, we verified that the attack is thwarted even when the synthetic graphs are generated from non differentially private instances of the model.

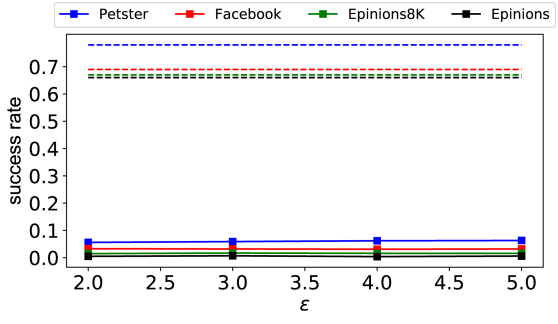
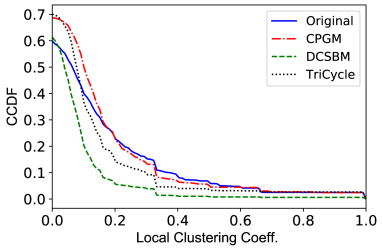


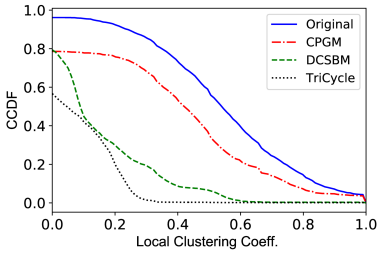
Fig. 3. Resistance against community-enhanced re-identification attack from [47].

F Distribution of local clustering coefficients

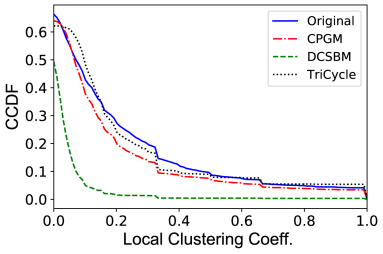
Fig. 4 shows the comparison of distributions of local clustering coefficients in terms of the *complementary cumulative distribution functions* (CCDF).



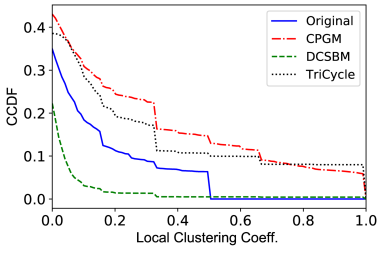
(a) Petster



(b) Facebook



(c) Epinions8K



(d) Epinions

Fig. 4. Comparison of edge generative models in terms of complementary cumulative distribution functions of local clustering coefficient distributions.